DisUnknown: Distilling Unknown Factors for Disentanglement Learning

Sitao Xiang[1,2], Yuming Gu[1,2], Pengda Xiang[1,2], Menglei Chai[3], Hao Li[2], Yajie Zhao[2], Mingming He*[2]
[1]University of Southern California, [2]USC Institute for Creative Technologies, [3]Snap Inc.
sitaoxia@usc.edu, {ygu, pxiang}@ict.usc.edu, mchai@snap.com, hao@hao-li.com,
{zhao, he}@ict.usc.edu

Abstract

Disentangling data into interpretable and independent factors is critical for controllable generation tasks. With the availability of labeled data, supervision can help enforce the separation of specific factors as expected. However, it is often expensive or even impossible to label every single factor to achieve fully-supervised disentanglement. In this paper, we adopt a general setting where all factors that are hard to label or identify are encapsulated as a single unknown factor. Under this setting, we propose a flexible weakly-supervised multi-factor disentanglement framework ***DisUnknown****, which* ***Dis****tills* ***Unknown*** *factors for enabling multi-conditional generation regarding both labeled and unknown factors. Specifically, a two-stage training approach is adopted to first disentangle the unknown factor with an effective and robust training method, and then train the final generator with the proper disentanglement of all labeled factors utilizing the unknown distillation. To demonstrate the generalization capacity and scalability of our method, we evaluate it on multiple benchmark datasets qualitatively and quantitatively and further apply it to various real-world applications on complicated datasets.*

1. Introduction

Disentanglement learning is the task of breaking down the tangled high-dimensional data variation into interpretable factors. In the desired disentangled representation, each dimension corresponds to a distinct factor of variables, such that when one factor changes, the others remain unaffected [3]. Disentanglement learning thus enables various downstream tasks such as transfer learning and few-shot learning, as well as challenging controllable image synthesis applications (*e.g.* [47, 14]).

With the availability of fully-labeled data, *supervised disentanglement* has seen much progress [29, 38, 15, 1, 14]. However, ground-truth labels are not always accessible, while even human labeling could be prohibitively expensive or inconsistent. Thus, fully-supervised approaches often have a hard time generalizing to common scenarios where labels are only partially available or even entirely missing. In light of this, *unsupervised disentanglement* approaches [10, 20, 27, 50, 42] have been proposed to address these challenges. However, most of them rely on the strong assumption that the target data is well-structured enough to be cleanly decoupled into explanatory and recoverable factors. And more importantly, there is no guarantee that these factors could be explicitly controlled with respect to the true intended semantics in specific manipulation scenarios. Therefore, *weakly-supervised disentanglement*, a nice mix of the best of both worlds, has recently become popular for more flexible learning [29, 45, 8, 17]. Unfortunately, although state-of-the-art performance is achieved on certain two-factor class-content disentanglement tasks [8, 17], most existing methods in this category are still unable to extract factor-aware latent representation, which is essential for manipulating individual factors especially when multiple ones are presented. In conclusion, no solution seems completely satisfactory yet on multi-factor disentanglement, due to the limited generalizability and insufficient performance.

In this paper, we propose a weakly-supervised multi-factor disentanglement learning framework, which handles arbitrary numbers of factors through explicit and near-orthogonal latent representation. Given that challenging factors that are hard to label or interpret exist in most tasks, the *key idea* to our approach is a general setting of N-factor disentanglement with $N-1$ factors labeled and a single factor unknown, where all the remaining task-irrelevant or difficult-to-label factors are flexibly encapsulated as one unknown factor. We find such a setting highly effective and practical in real scenarios. Take face motion retargeting as an example, facial expression could be a good candidate for the unknown factor since it is much more difficult to precisely label than others such as the identity and the pose. Thanks to its flexibility, our method naturally adapts to various tasks with varying domains (*e.g.* cartoon and real photos), data types (*e.g.* images, skeletons, and landmarks), in-

*Corresponding author.

tegrity (well-structured or in-the-wild), and label continuity (discrete or continuous).

To this end, our framework consists of two major stages: 1) *Unknown Factor Distillation* and 2) *Multi-Conditional Generation*. Specifically, we extract the unknown factor using an adversarial training method in the first stage, and then embed all labeled factors to the latent space as the second stage, which are used to condition the final generation. The core of our method lies in the joint adversarial training of factor encoders and discriminative classifiers, which explicitly disentangles unknown and known factors without introducing leakage between their disentangled representations.

The performance of our approach is extensively evaluated on several benchmark datasets, both qualitatively and quantitatively. Furthermore, we demonstrate the generalization capacity and practical robustness of the framework on multiple challenging tasks using complicated real-world datasets without any additional manual labeling effort.

Our contributions are: 1) A flexible weakly-supervised disentanglement learning framework that models data as a combination of labeled/unlabeled factors, which scales well to different datasets and benefits various challenging tasks; 2) A two-stage training architecture that explicitly learns disentangled representations for both labeled and unknown semantic factors, enabling mutual exclusive manipulation in the dimension of each factor; 3) A set of learning strategies to improve the effectiveness and robustness of adversarial training throughout our pipeline, which could potentially inspire future research; 4) State-of-the-art performance and wide range of practical uses on multiple challenging tasks including controllable image generation.

2. Related Work

Unsupervised Disentanglement has become the research focus because it does not require the access to the factors of variation. The pioneering work of InfoGAN [10], an information-theoretic extension to the Generative Adversarial Network framework [19], learns disentangled representations by maximizing the mutual information between the observations and a subset of latents. Considering its training instability and reduced diversity, the Variational Autoencoder (VAE)-based methods [20, 9, 30, 35, 27] are proposed for better performance and reconstruction quality by enforcing a factorized aggregated posterior on the latent space. However, these models are built on the assumption that the observations are independent and identically distributed in the datasets, thus successfully disentangled models may not be identified without any supervision [34]. Some task-specific unsupervised approaches disentangle two or more factors and achieve impressive results, such as image-to-image translation [21, 32, 43] and motion retargeting [49, 59]. These methods do learn disentangled representations, relying on specific categories [53, 48, 36, 59], clearly defined domains [21, 32, 43], or well-structured datasets with certain categories [50, 33]. In contrast, our method proposes a general framework, adapting to various tasks, domains, modalities and factor numbers.

Supervised Disentanglement requires strong supervision on specific factors of the data. These methods train a subset of the representations to match the known labels using supervised learning [44, 57]. With observed class labels only available for partial data, [22] and [40] propose semi-supervised VAE methods that learn disentangled representation. These supervised methods require large amounts of supervised data that would be expensive to acquire in practice. Although some methods can use synthetic data or data priors to provide full supervision [1, 14, 52], they are limited to processing domain-specific data such as human faces/bodies/hairstyles. Comparing to most supervised methods that only apply to specific tasks, what we propose is a general approach that applies to various applications.

Weakly-Supervised Disentanglement has been recently studied to build robust disentangled representations without requiring large amounts of data. Such weak supervision is provided as either known relations between the factors in different samples or ground truth labels of a subset of factors. To avoid explicitly labeling, some methods consider guiding disentanglement by matching pairs of data that share the same underlying factor [45, 29, 22, 4, 8]. By observing a subset of the ground truth factors, some methods perform distribution matching over data and observed factors and supervision is leveraged in style-content disentanglement with available labels for style only [28, 58, 26, 17]. Some of these methods may achieve state-of-the-art performance on certain class-content disentanglement tasks [8, 17], but they cannot ensure factor-aware latent representations for manipulating individual factors. The similar idea of a unified representation of labeled/unlabeled factors has emerged [16]. But we present a general disentanglement learning framework, which benefits various tasks.

3. Method

We propose a generic framework for weakly-supervised disentanglement learning and conditional generation. Instead of jointly training the whole system altogether, we take a two-stage approach. In the first stage, excluding all labeled factors, an encoder is trained to extract disentangled representation of the unknown factor from the input data. And in the second stage, with the unknown factor distilled, a conditional generative adversarial network is trained to embed the labeled data into the latent space, which allows independent control over each factor. By isolating the unknown factor from the labeled ones first, this two-stage training helps reduce the overall complexity of the task and improve the effectiveness of labeled factor disentanglement, as will be elaborated in the Training Strategy part in Stage II.

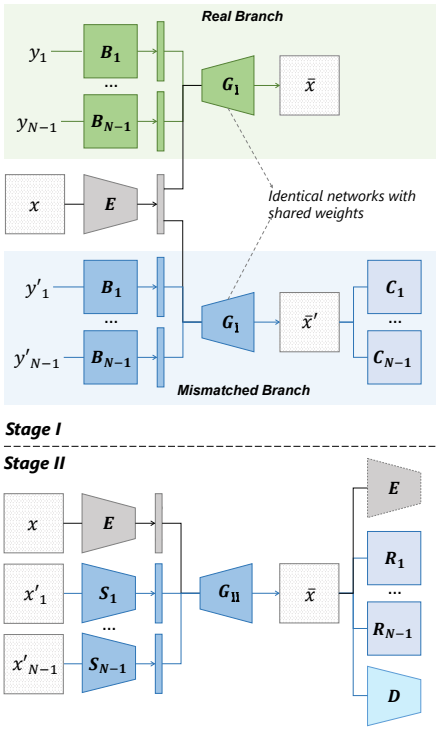


Figure 1: Illustration of our two-stage training architecture.

We note that Stage II is fully-supervised, in which missing labels for the unknown factor is provided by Stage I. Thus our method trivially covers the case where all factors are labeled, by dropping Stage I and using only Stage II.

3.1. Stage I: Unknown Factor Distillation

This stage trains an *unknown encoder* E that encodes the unknown factor completely and exclusively. It has two parallel branches in Figure 1 (*Stage I*), taking ground truth labels (in the *real branch*) and random labels (in the *mismatched branch*) of all known factors as input, respectively.

Specifically, let there be N factors, with the first $N-1$ ones labeled and the last one unlabeled. x is the training sample, $y = \{y_1, \dots, y_{N-1}\}$ are the associated *ground truth labels* and $y' = \{y'_1, \dots, y'_{N-1}\}$ are *random labels* chosen independently of x. E is the aforementioned *unknown encoder*, $B = \{B_1, \dots, B_{N-1}\}$ is a set of *label embedders*, both output normal distributions as in a VAE. G_I is the *Stage-I generator* that generates a sample $\overline{x}$ or $\overline{x}'$ for the real or mismatched branch, respectively, conditioned on E and B. $C = \{C_1, \dots, C_{N-1}\}$ is a set of *classifiers* that predicts the probability distribution of each factor from a generated sample. Both branches share network structures and weights. The loss functions of the two branches are summed. For now, we assume discrete labels, and discuss continuous-valued factors in the supplementary material.

Real branch: B map the ground truth labels y to normal distributions. We sample codes from these distributions and feed them to G_I, together with the distilled unknown factor from E, to generate the reconstructed sample $\overline{x}$.

Mismatched branch: By replacing the ground truth labels with random ones y'_i, G_I is asked to generate a mixed sample $\overline{x}'$. C_i predicts the ground truth label from the mixed sample, which indicates if any label information is leaked through E, since only E has the access to the ground truth factors in x. C are implemented as a single multi-class classifier that only branches at the last layer, and are trained with E in an adversarial manner.

Motivation. 1) In the real branch, by enforcing a reconstruction loss between the generated sample $\overline{x}$ and the original one x, E should include all information not covered by any labeled factor; 2) In the mismatched branch, by minimizing the accuracy of the classifiers C that are trying to predict the ground truth labels from the generated mixed sample $\overline{x}'$, E should exclude any information associated with the labeled factors to avoid label leaking.

Training Strategy. As a common problem of adversarial methods, jointly training the adversarial pair of E and C could be unstable. To improve the training robustness, we operate C on samples generated by G_I instead of codes sampled from the distributions produced by E (similar to [12]). This is because, without proper constraints, the distributions in the code space can fluctuate a lot in attempting to prevent the code from being classified. In contrast, with the reconstruction loss in the sample space, the distributions of the generated samples are close to the real ones, which avoids this kind of fluctuation.

As usual, the classifier C minimizes the *negative log-likelihood* (NLL). Let p be a vector representing the probability distribution for a particular factor and k be a class label whose probability is $p_{(k)}$, NLL is defined as:

$$\mathsf{NLL}(p, k) = -\ln p_{(k)}. \tag{1}$$

As the adversarial counterpart, the most obvious choice for the adversarial loss of E is to maximize the NLL loss. However, since NLL is not bounded when the probability $p_{(k)}$ is close to zero, E may prefer to focus on scoring very large NLL values on only a few samples rather than to make every output code equally unclassifiable. Therefore, instead of maximizing the NLL loss, we propose to minimize the *weighted negative log-unlikelihood loss* (NLU):

$$\mathsf{NLU}_q(p, k) = -\frac{1 - q_{(k)}}{q_{(k)}} \ln(1 - p_{(k)}), \tag{2}$$

where q are the reference distributions, which are always taken to be the actual class distributions in the training set for our purpose. In the supplementary material, we show how this definition of NLU loss is derived from the desired properties that it should be bounded, yield larger gradients on samples farther from equilibrium, and have the same equilibrium point as maximizing the NLL loss.

Full Objective. The full training objective on a single sample for Stage I is formulated as:

$$(\mu, \sigma^2) = E(x), \quad e \sim \mathcal{N}(\mu, \mathrm{diag}(\sigma^2)), \tag{3a}$$
$$(\alpha_i, \beta_i^2) = B_i(y_i), \quad b_i \sim \mathcal{N}(\alpha_i, \mathrm{diag}(\beta_i^2)), \tag{3b}$$
$$(\alpha_i', (\beta_i')^2) = B_i(y_i'), \quad b_i' \sim \mathcal{N}(\alpha_i', \mathrm{diag}((\beta_i')^2)), \tag{3c}$$
$$\overline{x} = G_{\mathrm{I}}(e, b_1, \ldots, b_{N-1}), \tag{3d}$$
$$\overline{x}' = G_{\mathrm{I}}(e, b_1', \ldots, b_{N-1}'),\ p_i = C_i(e, \overline{x}'), \tag{3e}$$
$$\mathcal{L}_C = \textstyle\sum_i \mathsf{NLL}(p_i, y_i), \tag{3f}$$
$$\begin{aligned}\mathcal{L}_{GEB} = {} & \mathsf{Rec}(x, \overline{x}) + \lambda_{\text{adv1}} \textstyle\sum_i \mathsf{NLU}_q(p_i, y_i) \\ & + \lambda_{\mathrm{KL}} D_{\mathrm{KL}}(\mathcal{N}(\mu, \mathrm{diag}(\sigma)) || \mathcal{N}(\mathbf{0}, I)) \\ & + \lambda_{\mathrm{KL}} \textstyle\sum_i D_{\mathrm{KL}}(\mathcal{N}(\alpha_i, \mathrm{diag}(\beta_i^2)) || \mathcal{N}(\mathbf{0}, I))).\end{aligned} \tag{3g}$$

The square on the variance vectors σ^2, β_i^2 and $(\beta_i')^2$ are per-element. $\mathsf{Rec}(x, \overline{x})$ is the reconstruction loss function, which is the mean squared error $||x - \overline{x}||^2$ in our experiments. D_{KL} is the KL-divergence. C are trained in the mismatched branch to minimize $\mathcal{L}_C$, averaged over all samples. E, B, and G_{I} jointly minimize $\mathcal{L}_{GEB}$.

3.2. Stage II: Multi-Conditional Generation

With the unknown factor distilled in Stage I, this second stage trains encoders S for labeled factors to extract the disentangled representations from the input samples. The final multi-conditional generator G_Π accepts conditions for both labeled and unknown factors, and ensures that varying one factor would not affect others in the generated output.

In this stage, as shown in Figure 1 (Stage II), the conditions of the unknown and labeled factors come from training samples x and $\{x_1', \ldots, x_{N-1}'\}$ respectively, all chosen independently. Each S_i of the *labeled-factor encoders* $S = \{S_1, \ldots, S_{N-1}\}$ computes the code for labeled factor i from x_i', while the *unknown encoder* E, pre-trained in Stage I, computes the unknown factor code from x. The *Stage-II generator* G_Π generates a sample $\overline{x}$ conditioned on all the codes (Eq. 5c). On $\overline{x}$, a set of *discriminative classifiers* $R = \{R_1, \ldots, R_{N-1}\}$ are trained to enforce the independent controllability of the labeled factor codes, and the pre-trained E is adopted to ensure the consistency of the unknown factor. In addition, a *discriminator* D is applied to ensure the realism of generated samples, as in GAN.

Motivation. Trained on random combinations of input samples, the generator G_Π is asked to synthesis a new sample with each factor conditioned by encodings from independent sources. Each classifiers R_i enforces that factor i of $\overline{x}$ is completely and solely controlled by x_i', and by choosing each x_i' randomly and independently we ensure that S_i is the only encoder that can consistently compute factor i of x_i'. The discriminator D makes the distribution of generated samples and real data indistinguishable globally.

Training Strategy. Most previous class-conditional GANs differ on how the generated sample is treated by the classifiers. Their classifiers are trained to correctly label the generated sample [41] or to be uncertain about the task [51]. But we go the opposite way: in addition to the NLL loss (Eq. 5e) for classifying the training sample x to the correct labels, our discriminative classifiers R are specifically trained to *not* classify the generated sample $\overline{x}$ correctly, by adding the *unweighted* NLU loss:

$$\mathsf{NLU}(p, k) = -\ln(1 - p_{(k)}). \tag{4}$$

Its rationale is that a conventional classifier oblivious to the generated samples tends to only learn just enough to distinguish one class from the others, which is insufficient to define the full characteristics of that class. However, if we ask the classifier to identify a generated sample as being in the wrong class, in order to tell real and generated samples apart it would be encouraged to gain a more complete understanding of each class.

G_Π and S are jointly trained to ensure that the generated sample $\overline{x}$ is classified to the same labels as the inputs $\{x_1', \ldots, x_{N-1}'\}$ (the NLL term in Eq. 5g).

Meanwhile, to enforce that the unlabeled factor is consistently controlled by the code from E, we minimize the distance between the encodings of the generated sample $\overline{x}$ and the input x, using the fixed E (square error term in Eq. 5g). This further explains why E must be trained in a separate stage from the rest of the system: E is used both for providing the input to the generator and for re-encoding the output to compare against the input. If E is allowed to be updated while this distance is being minimized, it could collapse to a state where it encodes everything to a zero vector.

As for the discriminator D, we use LSGAN loss functions [37] (Eq. 5f and the D term in Eq. 5g).

Full Objective. Similar to Stage I, the full training objective on a single sample for Stage II is formulated as:

$$(\mu, \sigma^2) = E(x), \quad e \sim \mathcal{N}(\mu, \mathrm{diag}(\sigma^2)), \tag{5a}$$
$$(\alpha_i', (\beta_i')^2) = S_i(x_i'), \quad s_i' \sim \mathcal{N}(\alpha_i', \mathrm{diag}((\beta_i')^2)), \tag{5b}$$
$$\overline{x} = G_\Pi(e, s_1', \ldots, s_{N-1}'), \quad (\overline{\mu}, \overline{\sigma}^2) = E(\overline{x}), \tag{5c}$$
$$p_i = R_i(x), \quad p_i' = R_i(\overline{x}), \tag{5d}$$
$$\mathcal{L}_R = \textstyle\sum_i (\mathsf{NLL}(p_i, y_i) + \mathsf{NLU}(p_i', y_i')), \tag{5e}$$
$$\mathcal{L}_D = (D(x) - 1)^2 + (D(\overline{x}) + 1)^2, \tag{5f}$$
$$\begin{aligned}\mathcal{L}_{GS} = {} & ||\overline{\mu} - \mu||^2 \\ & + \lambda_{\text{adv2}}(D(\overline{x})^2 + \textstyle\sum_i \mathsf{NLL}(p_i', y_i')) \\ & + \lambda_{\mathrm{KL}} \textstyle\sum_i D_{\mathrm{KL}}(\mathcal{N}(\alpha_i', \mathrm{diag}((\beta_i')^2)) || \mathcal{N}(\mathbf{0}, I)).\end{aligned} \tag{5g}$$

Note that while a total of N input samples are required to generate one sample, in practice this can be efficiently done by computing all factor codes for a whole batch and combining them randomly for generation. Classification labels are permuted accordingly. The classifiers R minimize $\mathcal{L}_R$,

Table 1: Unknown consistency ratios on *3D Shapes* with different unknown factors, w/ and w/o distillation.

Unknown Factor	w/ Distillation	w/o Distillation
Floor hue	100.00%	63.42%
Wall hue	100.00%	55.63%
Object hue	100.00%	68.76%

the discriminator D minimizes $\mathcal{L}_D$, and the generator G and encoders S jointly minimize $\mathcal{L}_{GS}$.

3.3. Implementation Details

For maximum generality we do not favor any specific network architecture. In all our experiments, encoders and generators consist of 3, 4, or 5 stride-2 convolutions for datasets with image sizes of 28, 64, or 128, respectively, followed by 3 fully-connected layers. Discriminators and classifiers have the same convolutional layers but only one fully-connected layer. The convolution feature map depth starts from 32 and doubles after each convolution but does not exceed 256. Fully-connected layers have 512 features.

4. Experiments

4.1. Datasets and Metrics

Datasets. We conduct evaluation experiments on four benchmark datasets: *MNIST* [31], *Fashion-MNIST* (*F-MNIST*) [56], *3D Chairs* [2], and *3D Shapes* [5]. For *MNIST* and *F-MNIST*, we use the standard training/testing split. For *3D Chairs* and *3D Shapes*, we randomly hold out 10% of all images for testing and use the rest for training. In *MNIST* and *F-MNIST*, we take *class* as the labeled factor since only it has labels available. In *3D Chairs* which contains three factors, i.e. *model*, *elevation*, and *azimuth*, we combine *elevation* and *azimuth* in to a single unknown factor of *rotation*. In *3D Shapes* which is fully defined by six labeled factors, i.e. *floor hue*, *wall hue*, *object hue*, *scale*, *shape*, and *orientation*, we select one or more factors as labeled and merge the remaining ones into the unknown factor to train various models for our empirical study.

Metrics. We evaluate the disentanglement performance by computing the Mutual Information Gap (MIG) [9] of the encoders. Since factors may contain more than one dimension, the mutual information of each factor is defined as the largest one over all dimensions. Then the MIG is computed as the gap of mutual information between the top two factors. Higher MIGs indicate better disentanglement quality.

4.2. Empirical Study

We empirically study how unknown distillation contributes to the disentanglement of labeled factors and enables control over the unknown factor.

Table 2: Labeled consistency ratios and MIG scores on *3D Shapes* with the unknown factor merged from varying numbers of factors. Zero unknown means fully-supervised.

# Unknown	Ratio	MIG ↑
0	100.00%	0.9501
1	100.00%	0.9555
2	100.00%	0.9733
3	100.00%	0.9718
4	100.00%	0.9393
5	100.00%	0.9868

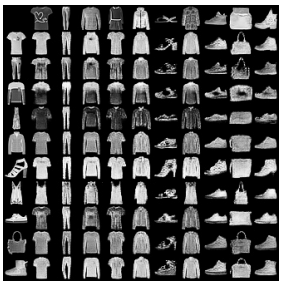
Table 3: Mean squared error (MSE) and MIG scores on *3D Shapes* with different unknown factor.

Unknown Factor	MSE ↓	MIG ↑
Floor hue	0.00049	0.9607
Wall hue	0.00063	0.9825
Object hue	0.00074	0.9766
Scale	0.00062	0.9411
Shape	0.00064	0.9637
Orientation	0.00064	0.9537

Necessity of the Unknown Factor. Without the unknown distillation, there is no guarantee that the features represented by the unknown factor remain fixed when altering any labeled ones. To compare, we modify Stage II by replacing the unknown factor code encoded by E with Gaussian noise and removing the feature matching loss $||\overline{\mu}-\mu||^2$ (Eq. 5g), and train three models on *3D Shapes*, with each selecting *floor hue*, *wall hue*, and *object hue* as the unknown factor, respectively. We generate images using the same random code for the unknown factor and independently-sampled random codes for all labeled factors, and then calculate the ratio of results sharing the same unknown feature, namely *consistency ratio*. Due to the simplicity of *3D Shapes*, these three features can be reliably computed by taking the colors at fixed pixel coordinates. Two colors are considered the same if their $L2$ RGB distance is less than half of the mean distance between two adjacent hue samples in the dataset. We generate 10,000 images for each network, and show the results in Table 1. As can be seen, all ratios reach 100% with distillation, meaning the unknown factor remains unchanged for all test samples. Note that MIGs are not measured here because the disentanglement performance among labeled factors is generally not affected.

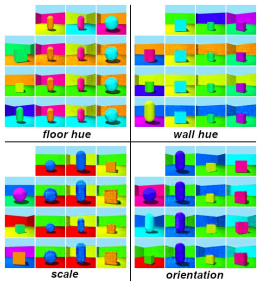
Scope of the Unknown Factor. In our setting, if there is more than one unknown factor, all these factors will be treated as a whole without individual controllability. However, we can still ensure that the unknown factors are isolated from the labeled ones, and the disentanglement performance of the labeled factors will not be influenced. To verify this, we train six models on *3D Shapes*: starting all

(a) MNIST/ class/ style



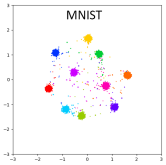
(b) F-MNIST/ class/ style

(c) 3D Chairs/ model/ rotation

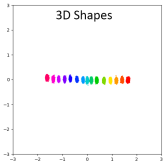


(d) 3D Shapes/ noted/ others

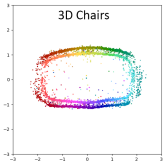
Figure 2: Generated samples on different datasets. The top row and the leftmost column are the input conditions for the labeled and the unknown factors, respectively, annotated as *dataset / labeled / unknown* in the sub-captions.



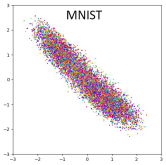
(a) style/class/class



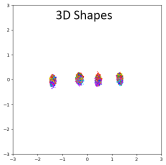
(b) ori./ori./ori.



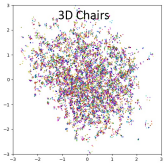
(c) rot./rot./rot.



(d) style/style/class



(e) shape/shape/floor



(f) rot./model/rot.

Figure 3: Visualizing the disentanglement with test sample distributions. The sub-caption of each figure represents: *unknown factor/ encoding factor/ coloring factor*.

factors labeled, we successively merge *floor hue*, *orientation*, *wall hue*, *scale*, and *shape* into the unknown factor, with *object hue* being the last labeled factor at the end. We measure the consistency ratios as introduced in *Necessity of the Unknown Factor* and MIG scores on *object hue* only in Table 2. Note that all MIG scores are quite close to the upper bound of 1, suggesting good disentanglement quality.

Choice of the Unknown Factor. We also study the robustness of our method by choosing different factors as the unknown one on *3D Shapes*. The MSE and MIG results, reflecting the consistent performance of reconstruction and disentanglement, respectively, are shown in Table 3.

4.3. Results and Visualizations

To demonstrate the quality of our multi-conditional generator, we plot the generated samples with factors controlled by random references on the benchmark datasets. As shown in Figure 2, our method accurately encodes both known (the top row) and unknown (the leftmost column) factors and uses them to independently control the generation.

We also illustrate the disentanglement quality by visualizing the test sample distributions in the code spaces in Figure 3. For each figure, we pick one encoding factor and one coloring factor from all factors, where both factors may or may not be the same. To draw each test sample on the 2D visualization, we generate the 2D position with the encoding factor and the color with the coloring factor. Specifically, we get its factor code using the encoder corresponding to the encoding factor and project it to 2D by selecting two dimensions with the largest variance. Then we draw a point on that 2D projection using the color mapped to its label of the coloring factor. The indication of good disentanglement is that colors should be clearly separated when the encoding and coloring factors are identical, but entirely mixed with no color pattern or bias when they are different.

4.4. Comparisons

We compare our approach against the state-of-the-art, including unsupervised [20, 27, 9] and weakly-supervised methods [8, 17]. The weakly-supervised methods are run under the same setting as ours where only one factor is labeled for *MNIST*, *F-MNIST*, and *3D Chairs*. Suggested hyperparameters are used to train these models: $\beta = 4$ for [20]; $\gamma = 10$ on *MNIST* and *F-MNIST*, and $\gamma = 3.2$ on *3D Chairs* for [27]; $\beta = 6$ for [9]; and $\beta = 10$ for [8].

From the results in Table 4, our method achieves substantially higher MIG scores than other methods on all datasets. Since the unsupervised methods [20, 27, 9] are trained without any supervision, comparing with them is somewhat unfair. Nevertheless, this emphasizes the importance of supervision in the disentanglement tasks, which is also reflected by the observation that the weakly-supervised methods consistently outperform the unsupervised ones.

We show a qualitative comparison in Figure 4 which rotates the *3D Chairs* images via traversing the latent code that depicts the azimuth rotation. The unsupervised methods [20, 27, 9] can smoothly change the orientation but fail to preserve the original style (*e.g.* shape, color, etc.). Among the weakly-supervised methods, [8] suffers from over-blurriness, while [17] cannot consistently control the orientation. Instead, our method is capable of handling var-

Table 4: The MIG scores of different disentanglement methods computed on the benchmark datasets.

Dataset	Unsupervised			Weakly-Supervised		
	[20]	[27]	[9]	[8]	[17]	Ours
MNIST	0.279	0.071	0.568	0.760	0.582	**0.978**
F-MNIST	0.105	0.043	0.111	0.630	0.539	**0.874**
3D Chairs	0.031	0.098	0.115	0.212	0.284	**0.404**

Figure 4: The rotation manipulation comparison on *3D Chairs* by uniformly sampling the latent codes depicting the azimuth rotation. The leftmost column shows the inputs.

ious chair styles and orientations, and achieves better generation quality with the original styles well preserved. Moreover, both weakly-supervised methods are limited to two-factor class-content disentanglement, but our approach is a more flexible multi-factor framework that supports factor-aware latent representation for each individual factor.

5. Downstream Tasks

Portrait Relighting. We train the network on the dataset combining celebA-HQ [23] and FFHQ [24] by treating the lighting as the labeled factor and the remaining content as unknown. Here, lighting is represented by second-order spherical harmonics coefficients for RGB and estimated with [25, 6]. Figure 5 shows our portrait relighting results.

Anime Style Transfer. We train the network on a custom dataset of $106,814$ anime portrait images drawn by $1,139$ artists collected online. The labeled factor is the artists' identity, which is used as the proxy for style. The unlabeled factor is interpreted as the content of the subject. Figure 6 shows our results on transferring style between different anime portrait illustrations, with comparisons to StarGAN [11] in multi-domain translation and the original Neural Style Transfer [18]. Our method achieves better results with styles more faithful to the examples.

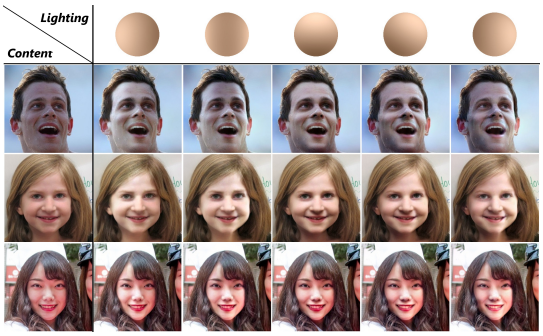


Figure 5: **Portrait relighting.** The top row shows various environment lightings mapped on a sphere. The leftmost column shows input images, and to the right are the re-lit results conditioned by the lightings in the same column.

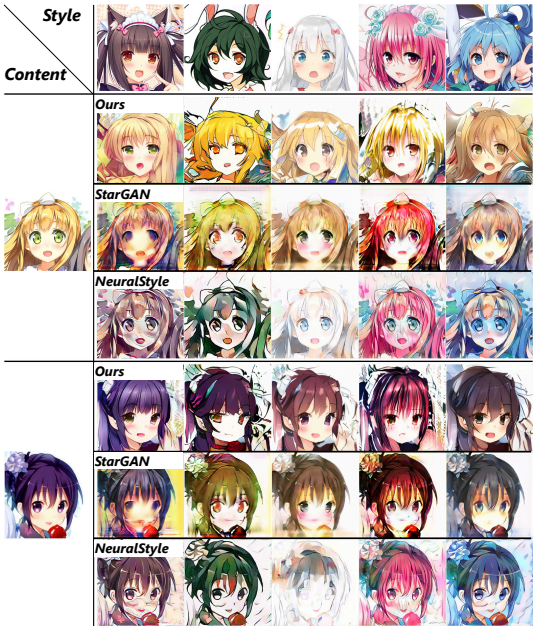


Figure 6: **Anime style transfer.** Each column is conditioned by the example style at the top row. In each group with three rows, the leftmost image is the content and the results are shown to the right. From top to bottom: our method, StarGAN [11], and Neural Style Transfer [18].

Landmark-Based Face Reenactment. We train our disentanglement network on facial landmark coordinates. After the new landmarks are synthesized with our generator, the output face images are translated from the rasterized landmarks using the image translation network (*e.g.* [54], [55]).

(a) Fix identity and pose, change facial expression.

(b) Fix identity and facial expression, change pose.

Figure 7: **Face reenactment with expression/pose control**. In each sub-figure, the leftmost column provides the identity and the pose/expression, and the top row provides the expression/pose. The reenactment results are generated with factors conditioned by these inputs.

Figure 8: **Face reenactment with factors from different sources**. The first three rows provide the identity, pose, and expression, respectively. The fourth row shows the results.

We use FD-GAN in [55] for one-shot image translation. The labeled factors are the identity and the head pose, where the pose is represented by Euler angles, estimated from the landmarks. The unlabeled factor is the facial expression. We train the network on VoxCeleb2 [13]. Figure 7-8 show our face reenactment results with various controls, including editing a single factor (expression/pose) (Figure 7) and mixing all three factors from different sources (Figure 8).

Skeleton-Based Body Motion Retargeting. We extract 2D joint coordinates from the driving videos and the actor images. The motion of the driving skeleton and the identity of the actor skeleton are combined to synthesize the target skeleton, with motion as the unknown factor. The images are generated using skeleton-guided synthesis (*e.g.* [46], [7]). Figure 9 shows the motion retargeting results on real images trained on Mixamo [39], which demonstrate promising disentanglement between identity and motion.

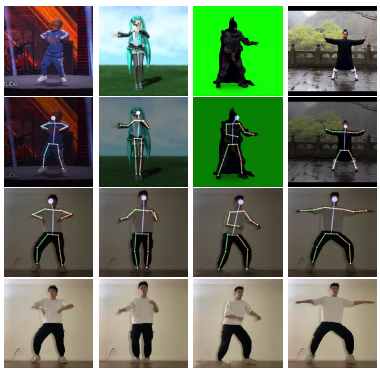


Figure 9: **Body motion retargeting.** From top to bottom in each column: input source frame, extracted source skeleton, transformed skeleton, and generated frame using [46].

6. Conclusion

We propose *DisUnknown*, a weakly-supervised multi-factor disentanglement learning framework. By distilling unknown factors, it enables independent control over each factor for multi-conditional generation. Our approach achieves state-of-the-art performance compared to existing unsupervised and weakly-supervised methods on multiple benchmark datasets. We further demonstrate its generalization capacity through various downstream tasks. Moreover, as a general framework, it can easily carry over to other modalities (*e.g.* text, audio) and help improve the stability of other tasks with our adversarial training strategies.

Acknowledgements

This research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-20-2-0053, and sponsored by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, and in part by the ONR YIP grant N00014-17-S-FO14. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation. Sitao Xiang wishes to dedicate this work to Sayori, his favorite illustrator, who has always been his inspiration.

References

[1] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *ACM Trans. Graph.*, 38(4):75:1–75:14, 2019.

[2] Mathieu Aubry, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic. Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models. In *CVPR 2014*, pages 3762–3769, 2014.

[3] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[4] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI 2018*, pages 2095–2102, 2018.

[5] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[6] Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph.*, 34(6):204:1–204:10, 2015.

[7] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. In *ICCV 2019*, pages 5932–5941, 2019.

[8] Junxiang Chen and Kayhan Batmanghelich. Weakly supervised disentanglement by pairwise similarities. In *AAAI 2020*, pages 3495–3502, 2020.

[9] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS 2018*, pages 2615–2625, 2018.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NeurIPS 2016*, pages 2172–2180, 2016.

[11] Yunjey Choi, Min-Je 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 2018*, pages 8789–8797, 2018.

[12] Ju-Chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-Shan Lee. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. In *Interspeech 2018*, pages 501–505, 2018.

[13] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep Speaker Recognition. In *Interspeech 2018*, pages 1086–1090, 2018.

[14] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and Controllable Face Image Generation via 3D Imitative-Contrastive Learning. In *CVPR 2020*, pages 5153–5162, 2020.

[15] Zunlei Feng, Xinchao Wang, Chenglong Ke, Anxiang Zeng, Dacheng Tao, and Mingli Song. Dual swap disentangling. In *NeurIPS 2018*, pages 5898–5908, 2018.

[16] Zunlei Feng, Zhenyun Yu, Yongcheng Jing, Sai Wu, Mingli Song, Yezhou Yang, and Junxiao Jiang. Interpretable partitioned embedding for intelligent multi-item fashion outfit composition. *ACM Trans. Multim. Comput. Commun. Appl.*, 15(2s):61:1–61:20, 2019.

[17] Aviv Gabbay and Yedid Hoshen. Demystifying inter-class disentanglement. In *ICLR 2020*, 2020.

[18] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR 2016*, pages 2414–2423, 2016.

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

[20] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR 2017*, 2017.

[21] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV 2018*, volume 11207, pages 179–196, 2018.

[22] Theofanis Karaletsos, Serge J. Belongie, and Gunnar Rätsch. When crowds hold privileges: Bayesian unsupervised representation learning with oracle constraints. In *ICLR 2016*, 2016.

[23] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR 2018*, 2018.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR 2019*, pages 4401–4410, 2019.

[25] Ira Kemelmacher-Shlizerman and Ronen Basri. 3D Face Reconstruction from a Single Image Using a Single Reference Face Shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):394–405, 2011.

[26] Bo-Kyeong Kim, Sungjin Park, Geon-min Kim, and Soo-Young Lee. Semi-supervised disentanglement with independent vector variational autoencoders. *CoRR*, abs/2003.06581, 2020.

[27] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML 2018*, volume 80, pages 2654–2663, 2018.

[28] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *NeurIPS 2014*, pages 3581–3589, 2014.

[29] Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NeurIPS 2015*, pages 2539–2547, 2015.

[30] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *ICLR 2018*, 2018.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang.

DRIT++: diverse image-to-image translation via disentangled representations. *Int. J. Comput. Vis.*, 128(10):2402–2417, 2020.

[33] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. MixNMatch: Multifactor Disentanglement and Encoding for Conditional Image Generation. In *CVPR 2020*, pages 8036–8045, 2020.

[34] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML 2019*, volume 97, pages 4114–4124, 2019.

[35] Romain Lopez, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. In *NeurIPS 2018*, pages 6117–6128, 2018.

[36] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR 2019*, pages 10955–10964, 2019.

[37] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV 2017*, pages 2813–2821, 2017.

[38] Michaël Mathieu, Junbo Jake Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. Disentangling factors of variation in deep representations using adversarial training. *CoRR*, abs/1611.03383, 2016.

[39] Mixamo. Mixamo. `https://www.mixamo.com/`.

[40] Siddharth Narayanaswamy, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah D. Goodman, Pushmeet Kohli, Frank D. Wood, and Philip H. S. Torr. Learning disentangled representations with semi-supervised deep generative models. In *NeurIPS 2017*, pages 5925–5935, 2017.

[41] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *ICML 2017*, volume 70, pages 2642–2651, 2017.

[42] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. In *ICLR 2019*, 2019.

[43] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. *CoRR*, abs/2001.05017, 2020.

[44] Scott E. Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML 2014*, volume 32, pages 1431–1439, 2014.

[45] Scott E. Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *NeurIPS 2015*, pages 1252–1260, 2015.

[46] Jian Ren, Menglei Chai, Sergey Tulyakov, Chen Fang, Xiaohui Shen, and Jianchao Yang. Human motion transfer from poses in the wild. In *ECCV 2020 Workshops*, volume 12537, pages 262–279, 2020.

[47] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *CoRR*, abs/2005.09635, 2020.

[48] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Güler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV 2018*, volume 11214, pages 664–680, 2018.

[49] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS 2019*, pages 7135–7145, 2019.

[50] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: Unsupervised Hierarchical Disentanglement for Fine-Grained Object Generation and Discovery. In *CVPR 2019*, pages 6490–6499, 2019.

[51] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR 2016*, 2016.

[52] Zhentao Tan, Menglei Chai, Dongdong Chen, Jing Liao, Qi Chu, Lu Yuan, Sergey Tulyakov, and Nenghai Yu. MichiGAN: multi-input-conditioned hair image generation for portrait editing. *ACM Trans. Graph.*, 39(4):95, 2020.

[53] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR 2017*, pages 1283–1292, 2017.

[54] 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 *CVPR 2018*, pages 8798–8807, 2018.

[55] Sitao Xiang, Yuming Gu, Pengda Xiang, Mingming He, Koki Nagano, Haiwei Chen, and Hao Li. One-shot identity-preserving portrait reenactment. *CoRR*, abs/2004.12452, 2020.

[56] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747, 2017.

[57] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. DNA-GAN: learning disentangled representations from multi-attribute images. In *ICLR 2018*, 2018.

[58] Jimei Yang, Scott E. Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis. In *NeurIPS 2015*, pages 1099–1107, 2015.

[59] Zhuoqian Yang, Wentao Zhu, Wayne Wu, Chen Qian, Qiang Zhou, Bolei Zhou, and Chen Change Loy. TransMoMo: Invariance-Driven Unsupervised Video Motion Retargeting. In *CVPR 2020*, pages 5305–5314, 2020.