Contents

1 Introduction

Computer vision is the scientific endeavour to algorithmically understand patterns in images. Structures and processes in the physical world interact in complex ways to generate an image, the image acts as a mirror, in which these elements of the world are reflected and leave patterns. To recognize patterns in an image, effectively means, to use this mirror as a window to observe the reality lurking behind it, *i.e*. to measure the causal elements that contributed to the image generation. Typically, objects appear in an intricated interaction of many factors of variation. For example, given the object class of people, persons can vary in their visual appearance by clothing and skin color or in their geometric structure due to their pose or body physique. For articulated object classes the most prominent factors of variation are geometric shape and visual appearance. Disentangling these factors is a difficult problem, due to the intricated interplay of shape and appearance under articulation. The complexity enters, as a variation in shape is a change of the images domain rather than a change of its values [**?**]. Consider a person raising his arm: the color and texture of his pullover sleeve intrinsically does not change, but appears at a different location in the image. An efficient model for shape should cover all possible states of the object and preserve the local linkage to its intrinsic appearance.

1.1 Why disentangle causal factors?

On the one hand, there are pragmatic reasons to aim at extracting disentangled factors from images: to successfully transfer a representation between different tasks, typically only a few factors are relevant [**?**]. Efficient transfer and multi-task learning should account for this. On the other hand, learning to capture external mechanisms in appropriate internal representations, can be seen as a step to automate reasoning itself. Once disentangled, a factor can be manipulated individually to make a targeted change. This enables machines to reason about the world [**?**], by simulating changes to factors internally in their model of the world. Thought experiments like *"imagine, how ridiculous you would look, if you wore that hot pants"* are managable tasks for the human imagination, but are out of the league for currently used generative image models [**?**, **?**], that typically rely on uninterpretable vector spaces with entangled dimensions. Building imagination machines has been proposed as a goal for artificial intelligence research recently [**?**]. To imagine, is to manipulate of an internal model to generate internal images. In this sense, in the context of generative modelling, disentangling factors could as well lead the way from a science of images to a science of imagination.

Figure 1.1: The image captions are generated by a deep neural network (Neuraltalk2) [**?**]. "Common sense" understanding of psychological and physical entities in terms of a causal model is absent [**?**]. Instead, the neural network captions a correlating associations.

1.2 How not to disentangle.

Can machines tell a story? Observe your own mind, when viewing the images shown in Fig. 1.1; observe how the human mind immediately interprets and jumps to conclusions, tries to tell itself a story that explains an image, whereas the machine (in this case, NeuralTalk2 [**?**]), is comically descriptive in contrast. The missing "common sense" may be due to a missing causal reasoning, due to a missing disentangled causal representation of the world. But how to learn a disentangled representation from scratch, *i.e*. from raw image data? As we will find out , disentangling causal factors from raw image data, without any side information is impossible theoretically, and can only work based on statistical assumptions. Lets consider an abstract example to enter : Given an image dataset of human persons that has strong variation in the pose and in the appearance of the persons, how to find these two underlying axes of variation (pose, appearance)? Lets suppose the distribution of varation follows a two-dimensional Gaussian distribution, one dimension for pose, one for appearance. The learning algorithm has access to random samples from this distribution. An intelligent data compression algorithm will be able to fit a function from the images to the two-dimensional subspace which explains (by assumption in this example) the varition in the dataset. But are the two dimensions that the algorithms finds disentangled? No. In fact, a linear combination of pose and appearance and its orthogonal complement are equally valid. Just from observing a two-dimensional Gaussian, no meaning will be attached to the axes. In practice, this problem is often circumvented interpolating in the latent space afterwards and determining the axes of interest (here the pose or appearance axis). The meaning of pose and appearance as independent factors comes from the fact, that it is easily possible in the real world to change one factor without the other. A person moving without loosing clothes is a trivial example for that. In summary, on the basis of dataset statistics one cannot disentangle arbitrary causal factors. The information about how to select the axes, *i.e*. which factors separate, is not contained in raw data. Fitting a model to the data distribution, does in general not yield insights about how the data was generated.

1.3 How can humans disentangle?

The dichotomy between humans and machines is constructed, of course, since on a fundamental level humans are machines. But in this context the differing between humans and artificial machines shall refer to the current gap between human and machine learning performance. So, what characteristics or advantages does the human mind have, that are lacking in data-driven machine learning?

Priors. Whether acquired or inherited, certain inductive priors seem to guide the human learning in its early phases [**?**]. Archetypal knowledge of psychology , a universal grammar for language and causal intuitions on everyday physics are some of the cognitive priors, that could explain the intuitive psychology, the rapid language acquisition and the remarkable causal inference from limited amount of data.

Data. Not only quantity, but also quality of data. Machine learning on images is commonly posed as the task to learn from randomly sampled images from a data set. Humans do not perceive the world randomly. To humans, the world appears in a temporal sequence, which reveals how factors change and persevere across time. Instead of focussing on datasets with static images, sampled at random so that the images may have nothing to do with each other, algorithms should use video datasets and harness the rich temporal information.

Most importantly, humans interact with their environment. That means, humans know change, not only by observing change (as in a temporal sequence), but also by interacting directly: how elements of the world change is known by changing the world. Anyone, who has watched a human infant play, can affirm that the learning mind is obsessed with interaction. Whether consciously by performing controlled experiments or by subconscious cues : *Interaction is crucial for a learning mind.*

Models. Humans are able to imagine. That presupposes an internal model of the world, to which specific changes of representational factors can be applied. In machine learning, fitting neural network models as functions to approximate datasets has seen tremendous progress recently, to the point, that it is considered a solved problem . This progress is mainly due to the effectiveness of neural networks to fit high-dimensional functions. But a probabilistic model to a dataset, however complex and rich, is not a causal model. Even if one were to obtain a probabilistic model over all images the world (one could start with *e.g*. ImageNet), this would tell very little about the real-world (causal) relationships between objects.

What can we learn from these differences? An algorithm to understand the world: should contain useful *prior* assumptions to efficiently use *data* that contains the necessary causal relationships and interactions, to learn a useful *model* of the world.

1.4 How to disentangle?

change factor $\rightarrow$ image change equivariantly, leave others invariant $\rightarrow$ equivariance, invariance

change can be mimicked artificially Intelligent pattern recognition algorithms, fuelled

by sensory data as learning material alone, may ultimately drive the way to a full-blown artificial intelligence, reasoning about the world on its own. - That is the reasoning behind data-driven and assumptionless machine learning approaches that have conquered several research communities. A theoretical objection to driving-only-with-data comes from the causal literature: For an understanding of the world, an algorithm needs to model causal processes, that cause an image to be generated.

1.5 Contributions

This thesis makes two theses:

- *Hypothesis* i)*: Unsupervised learning of object shape benefits from abstracting away the shapes complement, namely the object appearance. Explaining away the appearance factor can be achieved by a disentangled generative modelling of both factors.*

- *Hypothesis* ii)*: Learning unsupervised disentanglement without any assumptions is fundamentally constrained. In accordance with the literature on causal learning ([**?**]), disentangling causal factors requires model assumptions and/or interactional data - instead of observational (raw) data.*

To address these hypotheses, we *explain*, *validate* and *evaluate* a method for unsupervised shape learning: *Unsupervised Part-wise Disentanglement of Shape and Appearance* developed by .

To *explain*, we give an overview over state-of-the-art unsupervised disentangling literature and situate the proposed method in relation to the literature. In particular, we carve out the necessary aspects of an approach for disentangling causal factors and analyze the current state of research in order to indicate future directions.

To *validate*, we show that the proposed method outperforms the state-of-the-art for unsupervised learning of object shape on miscellaneous datasets, featuring human and animal faces and bodies. We also contribute several self-made video datasets for disentangling human pose from appearance, for articulated animal motion and for articulated composite objects. We highlight the specific challenges of these datasets and elucidate how the proposed method tackles them.

To *evaluate*, we perform ablation studies on critical components of the method. In addition, we compare to a part-wise shape learning method which does make the goal of disentangling explicit. To show that the disentanglement is indeed achieved, we evaluate the disentanglement performance against a shape-supervised state-of-the-art disentanglement method and perform favorably.

In short, our results a big improvement upon the state-of-the-art in unsupervised object shape learning. This confirms the first hypothesis. To complement the learned shape in a generative process, object appearance is disentangled from shape. The achieved disentanglement confirms the second hypothesis.

2 Prerequisites on Learning Disentanglement

2.1 Learning from Data

Learning from data is commonly understood as the ability of algorithms to improve their performance on a task with experience accumulated from the observation of data [**?**]. The source of data is usually a dataset - set of data points $X = \{x_i | i \in \{1 \dots n\}\}$, which are sampled from a probability distribution $x_i \sim p(x)$.

2.1.1 Supervised

The term supervised learning denotes the task to learn a mapping from data points x_i to target labels y_i. A supervised algorithm has access to data-label pairs $(y_i, x_i) \sim p(y, x)$, in order to estimate the connection between data points and labels, either in form of a conditional probability $p(y|x)$, or in form of a deterministic function $y = f(x)$. The label y can be either discrete (*e.g.* information about an object class) or continuous (*e.g.* the location of an object part in an image). Recent advances, in particular the effectiveness of neural network models (cf. sec. 2.1.3) on big datasets, have led to huge progress on problems that can be formulated as regression or classification. That is why On many traditional computer vision problems, such as *e.g.* object recognition, image classification or human pose estimation, machines are now performing on a superhuman level; hence, many supervised problems are now considered to be essentially solved.
The Achilles' heel of supervised learning lies in the need for a viable supervision signal. To get labels it is usually requiring to manually annotate the data. The human effort in this is costly, error-prone and not scalable to the ever-growing vast amounts of raw data.

2.1.2 Unsupervised

Unsupervised learning is the endeavour to learn about structures and patterns in unlabelled data. In this paradigm, the learning algorithm has access to the samples of the data distribution $x \sim p(x)$. The task is usually framed as a form of density estimation, *i.e.* to model the entire distribution in a probabilitstic model (cf. sec. 2.2).

connection between unsupervised and supervised learning, cite dlb.

model-free vs model-based rigid enough to be useful, flexible enough to useful recently data-driven -> flexible

limits of unsupervised learning? how much prior modelling should be employ? -> as much as possible as long as it is good? (link post Inference)

modeling data distribution $p(y, x)$ sampling from distribution possible e.g. outlier detection where $p(x)$ has low probability

2.1.3 Artificial Neural Networks

Artificial neural networks are a powerful and flexible tool for function approximation. In their principles they are inspired by biological neural networks. An artificial network is model for a function $y = f(x)$ with vector input $x = \{x_i | i = 1 \dots n\}$ and vector output $y = \{y_j | j = 1 \dots m\}$:

$$\begin{aligned} h_j &= a(\sum_i w_{ji} x_i + b_i) \\ y_j &= a'(\sum_i w'_{ji} h_i + b'_i) \end{aligned} \tag{2.1}$$

, with weight matrices w, w', non-linear so-called activation functions a, a' (*e.g.* $a(x) = 0$ for $x < 0$, $a(x) = x$ for $x >= 0$) and bias vectors b, b'. The components h_j are called hidden units or neurons. Neural networks can also comprise multiple hidden layers via $h_j = a(\sum_i w_{ji} h_i + b_i)$. It can be shown theoretically, that in the limit of infinite hidden units h_j they can approximate any (continuous) function arbitrarily close [**?**] . In practice, however, networks with more that one layer, referred to as deep neural networks, seem to work better.

For processing image data, one constrains the weight matrices to be only locally connected and to share weights across locations to enforce translation invariance, resulting in *convolutional* neural networks.

longstanding model gained hype-status as working together optimization via gradient descent has proven successful (for deep networks called backpropagation) differentiable

2.2 Generative Models

> What I cannot create, I do not understand. - R. Feynman

Learning and understanding structure in data by being able to generate, is the rationale behind generative modelling. Generative models are mostly applied for unsupervised learning and can be contrasted to discriminative models. While discriminative models are used to model posterior conditionals $p(y|x)$ (*e.g.* for supervised learning (cf. sec. 2.1.1), generative models capture the complete data distribution $p(x)$ in an estimate $\hat{p}(x)$. Thus, after estimation, one can generate samples from this model $\hat{p}$, hence the name generative model. The currently predominant formulations for learning generative models are built on either autoencoding or adversarial formulations:

2.2.1 Autoencoding Formulations

An autoencoding model is learning by reconstructing samples of data, $\hat{x} = f(x)$. To enforce data compression (otherwise the identity function is a trivial solution of autoen-

coding) the function has an information bottleneck, namely an inferred latent code z of reduced dimension. The autoencoder is then the chain of an encoding function $z = e(x)$ and a decoding function $\hat{x} = d(z) = d(e(x))$.

Whereas the conventional autoencoder consists of deterministic mappings e, d, the *variational autoencoder* models the probability distribution $p(x)$. More specifically, it maximizes a lower bound to the logarithmic likelihood $\log p(x)$ of data x. This so-called variational lower bound $\mathcal{L}$ is given by:

$$\mathcal{L} = \mathbb{E}_{z\sim q(z|x)} \log p(x|z) - \mathrm{KL}(q(z|x)||p(z)) \tag{2.2}$$

Where z introduces latent variables, with a prior distribution $p(z)$. The approximation to the posterior $q(z|x)$ of the latent variables and the posterior of the data given the latent variables $p(x|z)$. If one wants to model the distributions with neural networks, one typically uses Gaussian distributions and lets the networks predict the parameters (mean μ and variance Σ) based on the image. In the current machine learning contexts, all functions (e, d) and or moments (μ, Σ) are modelled with neural networks.

2.2.2 Adversarial Formulations

Generative Adversarial Networks (GAN) consist of two neural networks competing in a zero-sum game. A generator network G is generating images based on a latent code z sampled from a distribution $p(z)$. The discriminator network D is a binary classifier with the task to classify an image as originating from the data distribution p_{data} or from the distribution produced by G. The loss function of G is the negative of the loss of D, such that one can formulate the optimization in a minmax form:

$$\min_D \max_G -\frac{1}{2}\mathbb{E}_{x\sim p_{data}}[\log D(x)] - \frac{1}{2}\mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{2.3}$$

The discriminator can also be interpreted as a learned similarity metric to measure the closeness of an image to the data distribution. [**?**]. The generator is then optimized to make the output indiscriminable from the data distribution.

There are many variants and extensions to this basic principle of learning with an adversarial task. For example, one can learn a discriminator on for a set of image patches [**?**].

2.3 Disentangling Representations

In supervised learning, a performance measure is naturally induced by the metric that is being optimized. In the unsupervised setting, judging the performance of a model is less straightforward. For example, when modelling an image domain, one could subjectively rate the quality of the generated image. But even for a qualitative assessment the question arises, how to rate the quality of the latent representation?

2.3.1 Learning Representations

> Disentangle as many factors as possible, discarding as little information about the data as is practical. - [**?**]

According to [**?**] a representation is useful, if it can be applied to many - in advance unknown - different tasks, while being trained on only one particular task. As the downstream tasks can be multifarious, the essential *information* should be contained in the representation. For some tasks only a subset of aspects of the data will be necessary, that is why *disentangled factors* make a representation particularly practical - so goes their reasoning.
The latent representation z learned by generative models captures the essential *information* of the data distribution. That is made sure by requiring the ability to generate samples from the original data distribution from it. How then to reach the second goal, the *disentanglement* of generative factors:

2.3.2 Disentangling as Equivariance and Invariance

The definition of factor by change static ... factors should represent elements of real world - change in element -> corresponding change in representational factor - leave other factors representing other elements invariant

Formally, this can be posed as an inference problem: a number of latent variables $\mathbf{z_1} \dots \mathbf{z_N}$ has interacted in certain ways to cause the existence of the observed image $\mathbf{x}$. An inference algorithm aims at recovering these latent variables from the observation, *i.e.* the image. These recoveries can be seen as estimates $\hat{\mathbf{z}}_\mathbf{i}$ for - or a representation of - the true latent variables $\mathbf{z_i}$. A graphical model of the process is shown in figure 2.1. A disentangled representation should then represent each causal element and its state independently: A change in the real causal element $\mathbf{z_i}$ should correspond to an equivalent change in the abstract representational factor $\hat{\mathbf{z}}_\mathbf{i}$, while leaving the other factors $\hat{\mathbf{z}}_\mathbf{j}, j \neq i$, that represent other causes, unchanged.

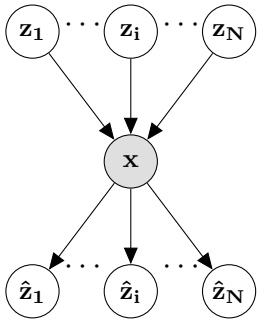


Figure 2.1: Disentangling causal factors means to infer an estimate - *i.e.* a representation - from an image

mathematically,.. $f \circ g(x) = ...$

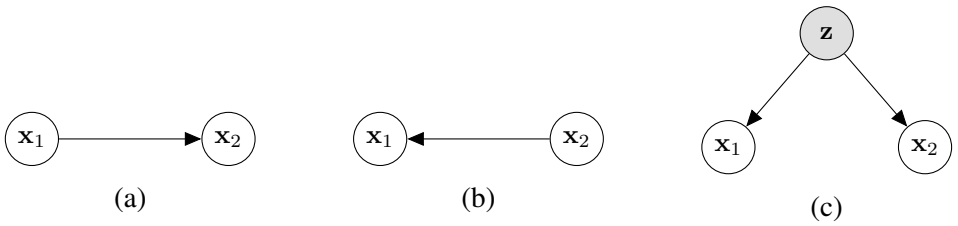


Figure 2.2: Correlation implies causation - if x_1 and x_2 correlate, a) x_1 may cause x_2, b) x_1 may be caused by x_2 or c) both are contingent on a latent cause z

2.4 Theoretical Impediments from Causality

As outlined earlier, the type of knowledge that can be gained by learning from "raw" data is limited. With raw data we mean data x sampled from a $p(x)$. so far fitting curve p(x) to data manifold what is missing to human-level intelligence? (cite lake 2016)

causal learning is a hard problem: instead of only learning statistical measures from data, model also needs to be learned ([**?**])

Hypothesis: disentangling factors = estimating causal factors -> needs causal for estimation of causal factors "raw data" insufficient -> need interventional data or model assumptions. we do both: 1. intervene with changes to an image which are assumed to change only one factor. 2. model the causal process of the image generation in the theme of analysis-by-synthesis

2.4.1 Causal Learning requires Interventions or Assumptions

What does the causality literature have to say? Statistic background $\rightarrow$ correlation is not causation. Reichenbachs principle [**?**] $\rightarrow$ barometer example: How to find out the causal connection between a barometer and the weather. Highly capable machine learning algorithm that learns only with access to an image dataset showing the barometer and the weather. -> will be able to capture the correlation between needle position and weather condition, but never understand causal direction, since it is not in the data. Imagine how a human would go about solving this problem. Having a mechanistic model of the world he could reason about the precise causal mechanism relating weather to humidity to needle position. - model of influences (humidity -> barometer) What if no prior knowledge? A child-level simple solution is to force the needle to move with a finger. The weather will not change. Hence causality has to go other way or third latent variable influencing both. - intervening: move barometer needle by hand -> no change in weather, hence causality has to go the other way, (example from [**?**]) There cannot be an abstract intelligence, which finds out about the world purely by observation. The intelligence has to interact with the world, it has to be in the world. before this becomes too philosophical infer causation from correlation RCT

lacking the tools to accurately estimate causality, researchers shied away from making causal statement. Developing machines with human-like abilities requires discovery and reasoning in terms of causal models. Recently (in the past 30 years), overshadowed by

the prominent success of data-driven deep learning, the field of causality has emerged to mathematical rigor.

- ladder of causation: association, intervention, counterfactual - current machine learning mostly on level of association (correlations estimated from "pure" data) -> purely data-driven approach can only go so far humans seem to have innate assumptions on coherence, causality, physics etc. introducing inductive biases

measure: p(x) assume causal model: p(x | a, s) want: p(s) and p(a)

encoding $p(s) = p(s|x)$ $p(a) = p(a|x) = p(a|s, x)$

decoding $p(x) = p(x|a, s)p(a)p(s)$

p(x| do(s), do(a))

example: Gaussian only with access to p(x) hopes to find factors p(a, b) = p(a) p(b) (InfoGAN, BetaVAE) what if not full-filled? two-dimensional Gaussian: axis x and y are independent factors. in general any superposition of x and y which is orthogonal, can be found imagine a perfect dimensionality reduction yielding a two-dimensional latent space one can find the axes that correlate most with human understanding of independent factors i.e. pose and appearance. But how can a machine find these axes automatically from raw data? it cant, neither can anyone (including humans) (Pearl). Humans know these factors are independent from observing that they can change independently e.g. from observing someone undressing or changing his pose (i.e. harnessing temporal information, with the assumption of temporal coherence) or by changing the factors themselves e.g. what happens to the image of me if I change my pullover? It can be proven mathematically (Pearl) that interventional data or at least certain (which) causal assumptions about the world are necessary to estimate certain quantities.

2.4.2 Interventions are Transformations

we harness intervention p(x| do(a), b) in computer vision an intervention is an image transformation if ..

2.4.3 Assumptions in Analysis-by-Synthesis

Inverse graphics Capsules, Tieleman make model as good as we can implementing as many assumptions as we can and only leave the rest to powerful model Synthesis known, analysis only indirectly by observing cognition

leaving synthesis to learning from scratch, can meet practical/computational limits *e.g.* convolutional neural networks better than fully connected neural models. But can also be ultimately impossible. Modelling synthesis explicitly with a causal model about image generation, by knowledge about the physical world enables answering interventional and counterfactional questions. (mathematically impossible to learn from ”pure” data alone)

2.5 Object Shape and Appearance

3 Analysis of Literature on Disentangling

3.1 Learning Object Shape

for estimating shape s from images x the task is $p(s|x)$ representation of shape can be landmarks

Disentangling generative factors definition model-free vs model-based approaches:

- model-based $\rightarrow$ more flexible, transferable, modular combination (like parts)

parts as regional attention (cite attention paper) parts/compositionality is key to creativity -> new combination of known parts

3.2 Analysis-by-Synthesis

Capsules, Tieleman make model as good as we can implementing as many assumptions as we can and only leave the rest to powerful model Synthesis known, analysis only indirectly by observing cognition

leaving synthesis to learning from scratch, can meet practical/computational limits *e.g.* convolutional neural networks better than fully connected neural models. But can also be ultimately impossible. Modelling synthesis explicitly with a causal model about image generation, by knowledge about the physical world enables answering interventional and counterfactional questions. (mathematically impossible to learn from "pure" data alone)

3.3 Disentangled Generative Models

Capturing essential information about data in a representation by being able to generate it is the rationale behind generative modelling. Currently the approaches in this direction are defined by adversarial [**?**] and autoencoding [**?**] model formulations. Recently, the endeavour for disentangling explanatory factors in the latent representation is being made explicit in the objective functions [**?**, **?**] of these models. So far, however, these attempts are limited to rigid objects without articulation and disentangle holistic image factors like illumination, object rotation or total shape and global appearance.

3.4 Disentangling Shape and Appearance

Factorizing an object representation into shape and appearance is a popular ansatz for representation learning. Recently, a lot of progress has been made in this direction by

conditioning generative models on shape information [**?**, **?**, **?**, **?**, **?**, **?**]. While most of them explain the object holistically, only few also introduce a factorization into parts [**?**, **?**]. In contrast to these shape-supervised approaches, we learn both shape and appearance without any supervision.

For unsupervised disentangling, several generative frameworks have been proposed [**?**, **?**, **?**, **?**, **?**, **?**]. However, these works use holistic models and show results on rather rigid objects and simple datasets, while we explicitly tackle strong articulation with a part-based formulation.

3.5 Part-based Representation Learning

Describing an object as an assembly of parts is a classical paradigm for learning an object representation in computer vision [**?**] with linkage to human perceptual theories [**?**]. What constitutes a part, is the defining question in this scheme. Defining parts by e.g. (*i*) visual/semantic features (object detection), or by (*ii*) geometric shape, behavior under (*iii*) viewpoint changes or (*iv*) object articulation, in general leads to a different partition of the object. Recently, most part learning has been employed for object recognition, such as in [**?**, **?**, **?**, **?**, **?**, **?**]. To solve such a discriminative task, parts will be based on the semantic connection to the object and can ignore their spatial arrangement and articulation of the object instance. Our method instead is driven by a generative process and aims at more generic modeling of the object as a whole. Hence, parts have to encode both spatial structure and visual appearance accurately. To our best knowledge unsupervised part learning and the proposed split in shape and appearance description for a part has only been used in pre-deep learning approaches [**?**, **?**, **?**].

3.6 Landmark Learning

There is an extensive literature on landmarks as compact representations of object structure. Most approaches, however, make use of manual landmark annotations as supervision signal [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**].

To tackle the problem without supervision, Thewlis *et al*. [**?**] proposed enforcing equivariance of landmark locations under artificial transformations of images. The equivariance idea had been formulated in earlier work [**?**] and has since been extended to learn a dense object-centric coordinate frame [**?**]. However, enforcing only equivariance encourages consistent landmarks at discriminable object locations, but disregards an explanatory coverage of the object.

Zhang *et al*. [**?**] addresses this issue: the equivariance task is supplemented by a reconstruction task in an autoencoder framework, which gives visual meaning to the landmarks. However, in contrast to our work, he does not disentangle shape and appearance of the object. Furthermore, his approach relies on a separation constraint in order to avoid the collapse of landmarks. This constraint results in an artificial, rather grid-like layout of landmarks, that does not scale to complex articulations.

Jakab *et al*. [**?**] proposes conditioning the generation on a landmark representation from another image. A global feature representation of one image is combined with the landmark positions of another image to reconstruct the latter. Instead of considering landmarks which only form a representation for spatial object structure, we factorize an object into local parts, each with its own shape *and* appearance description. Thus, parts are learned which meaningfully capture the variance of an object class in shape as well as in appearance.

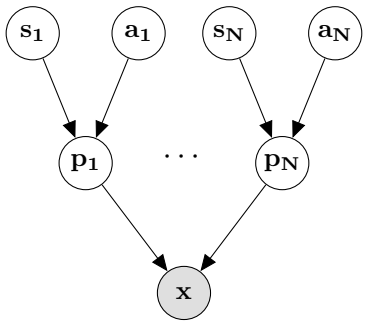
Additionally, and in contrast to all these works ([**?**, **?**, **?**]) we take the extend of parts into account, when formulating our equivariance constraint. Furthermore, we explicitly address the goal of disentangling shape and appearance on a part-based level by introducing invariance constraints.

4 Method

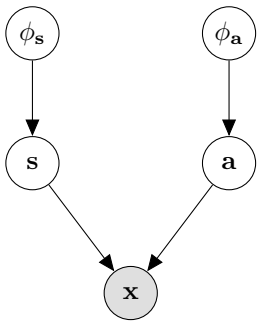
To capture an object in an abstract representation, we follow two key ideas: *(i)* disassembling the object into its constituent parts and *(ii)* disentangling spatial geometry (shape) from visual features (appearance). Hence, we model an object as a composition of parts, each part with a part appearance and a part shape, see Fig. **??**. The part shape should correspond to the area in the image where the part is located, whereas the part appearance is a feature descriptor for that area. The overall object representation is then the collection of part shapes and part appearances.
The disentanglement of shape and appearance can be enforced by demanding that shape is invariant under the transformation of appearance and vice versa. This is realized in a two-stream auto-encoding formulation. Here, an image is reconstructed from a combination of shape and appearance, with shape extracted from the appearance-transformed image and appearance from a shape-transformed image. Additionally, the part shape is tied to the location of the part in the image: an equivariance loss encourages that the part shape moves in unison with the part in the image. We implement these objectives into a loss framework, which is explained in sec. 4.1.
To assert a decomposition into independent local parts, we ensure their local modelling and treatment throughout the whole pipeline. This is highlighted when describing the architecture in sec. 4.2.

4.1 Framework

We want to represent an object in an image X. Let us denote the part shape with p_X and the part appearance with f_X. For an object with n parts, the overall shape is constituted by the collection of its part shapes $p_X = (p_X^1, ..., p_X^n)$, the same goes for the appearance $f_X = (f_X^1, ..., f_X^n)$. We model the part appearances as feature vectors, the part shapes are chosen to be scalar fields like the image itself. Thereby one can establish a direct correspondence of locations in the image to locations in the shape representation.
How do we disentangle the shape and appearance components in the representation? In general, a variation in shape will not affect appearance and vice versa. Thus, if we deliberately change shape without changing appearance, we can enforce the invariance of the appearance representation under such a change. We refer to these changes as shape transformations π, which, if applied to an image X, directly act on the underlying pixel space Λ. Along the same lines we can define appearance transformations ϕ, which act on the image itself. The shape should be invariant under change of appearance, conversely, the appearance should be invariant under change of shape. In addition, the shape should transform in the same manner as the image. That means the shape representation is as-



(a) Modelling an image of an object as a collection of parts, each with its own shape and appearance



(b) Implementing the do-operation with a transformation of factors

Figure 4.1

sumed to be equivariant under shape transformations. In summary:

$$f_{\pi(X)} = f_X \qquad \text{(invariance of appearance)}$$
$$p_{\phi(X)} = p_X \qquad \text{(invariance of shape)}$$
$$p_{\pi(X)} = \pi(p_X) \qquad \text{(equivariance of shape)}$$

Our method builds on the auto-encoding paradigm, with part shapes and part appearances assuming the role of the latent code. To incorporate these constraints into the loss of an auto-encoder, we reconstruct an image X not from the shape and appearance (f_X, p_X) determined from the original image X, but from appropriately transformed images $(f_{\pi(X)}, p_{\phi(X)})$. If the invariance constraints, as formulated above, are fullfilled, these transformations do not change the latent code. Thus, the loss implicitly enforces invariance. To obtain shape and appearance, we encode both $\phi(X)$ and $\pi(X)$ with an encoder E. And, after a recombination R (for details see sec. 4.2) to a latent image Z, a decoder D reconstructs the image. This configuration is depicted in Fig. 4.2, the reconstruction loss $\mathcal{L}_{\text{rec}}$ is as follows:

$$\mathcal{L}_{\text{rec}} = \|X - \mathrm{D}[\mathrm{R}(f_{\pi(X)}, p_{\phi(X)})]\| \tag{4.1}$$

Let us examine what this formulation means on the level of a single part: the part appearance f_X^i is extracted at locations in the spatially transformed image $p_{\pi(X)}^i$, but then used for reconstruction at the location in the original image p_X^i. For example in Fig. 4.2 the appearance of the arm will be extracted in a raised position, but then these features are used for reconstructing an arm in a lowered position. For this to succeed, firstly, the appearance features need to be sufficiently abstract. Secondly, part locations of the two images have to refer to the same part and track the location of it consistently. This part assignment consistency is an implicit way to improve equivariance under the shape transformations.

For a known shape transformation the equivariance of shape can also be encouraged explicitly with a loss. This has been used before in the context of unsupervised landmark

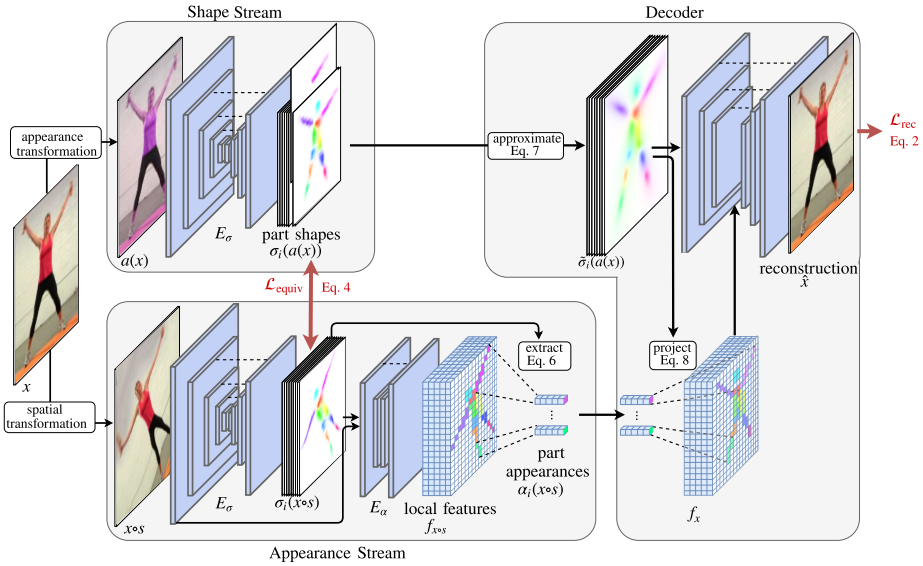


Figure 4.2: Encoder E encodes shape and appearance for two images $\pi(X)$ and $\phi(X)$, after recombination R of $(f_{\pi(X)}, p_{\phi(X)})$ into latent image Z, the decoder D reconstructs the image X.

learning by [**?**, **?**] as a point-wise loss on a part probability map, encouraging the exact location of a part to transform accordingly. In our case, the part shapes shall not encode probability, but instead the spatial extend of a part. In approximation, we want the first two moments (μ, Σ) to transform correctly. Thereby the extend and orientation of the parts is penalized in addition to the mere position.

$$\mathcal{L}^i_{\text{equiv}} = \mathcal{L}^i_\mu + \mathcal{L}^i_\Sigma \tag{4.2}$$

The overall loss objective is the sum of the reconstruction loss and the equivariance loss for all n parts:

$$\mathcal{L} = \sum_{i=1}^{n} \mathcal{L}^i_{\text{equiv}} + \mathcal{L}_{\text{rec}} \tag{4.3}$$

4.2 Architecture

The auto-encoding pipeline consists of three stages, namely: **encoding** both shape and appearance for each part, **recombining** this information meaningfully into a latent image

and **decoding** this latent image to reconstruct the image. The whole process is sketched in Fig. 4.2, the operations in more detail are visualized in Fig. **??**, **??**, **??**, **??**. Throughout the procedure we maintain the local correspondence between the representation and the image: We ensure a local appearance extraction in the encoding, a local synthesis in the recombining and a local usage of the latent image in the decoding. These architectural restrictions enable a disentangled part representation with the interpretation of a part as a localized entity.

4.2.1 Analysis

$(f, p|X)$[1] The encoding of shape and appearance given an image proceeds in two steps:
(i.) $(p|X)$: The part shapes are predicted given the image. To extract part shapes we use an hourglass[2] neural network: The input is an image X, the output a stack of n part shapes $s = \{p^i | i = 1, ..., n\}$.
(ii.) $(p|f, X)$: The part appearances $f = \{f^i | i = 1, ..., n\}$ are predicted given the image and the part shapes. Again we use an hourglass network, albeit shallower. The input is the original image concatenated with the stack of part shapes. The output is a feature stack F. A part appearance is obtained by averaging the feature stack with the a part shape: $f^i = \sum_{x \in \Lambda} A(x) \frac{p^i(x)}{\sum_{x' \in \Lambda} p^i(x')}$. Each f^i now describes the appearance of a part spatially localized by the part shape p^i.

4.2.2 Recombination of Factors

In the analysis-by-synthesis regime, once the object representation is successfully factorized, one can make assumptions on how the factors reunite to generate an image, following the knowledge and intuition about how objects give rise to images in the physical world. Firstly, we remerge shape and appearance into images of descriptors at the correct locations. For each part, appearance is multiplied with the corresponding shape, yielding n part feature images: $z^i(x) = p^i(x) \cdot f^i$. Secondly, we reassemble the object from its parts: the part feature images z^i are summarized by summing in a single image: $Z(x) = \sum_i \frac{z^i(x)}{1 + \sum_j z^j(x)}$. The result is an image of part feature descriptors located according to their corresponding part shape, which we call latent image Z.

[1] For a slim notation, we leave out the explicit reference to the generic input image X in this section: f, p, f^i, p^i refer to f_X, p_X, f^i_X, p^i_X.
[2] We utilize hourglass neural network models in both steps, as these models are able to preserve pixel-wise locality, but integrate information from multiple scales [**?**].

4.2.3 Synthesis

Finally, the latent image needs to be decoded to an image. This is done by a neural network decoder. The decoder architecture is modeled after the upsampling stream of a standard U-Net []. The latent image is scaled to different scales and inserted, after each layer, in addition to the part shapes . As before, the crucial property of the parts that needs to be conserved is their local direct correspondence to the image. On the one hand, one needs to assure, that the receptive field of the neurons does not extend to the full image, in order to thwart a complex non-local interaction of part information. This is why we use only half of a U-Net instead of a complete U-Net or an hourglass architecture. On the other hand, it is essential to regularize the information already before passing it to the decoder. Keeping in mind that the part shape should be of rather simple geometry, we introduce a differentiable information bottlenecks, in order to prevent the shape from being scattered over the object. It is an approximation of the part shape as $\hat{p}^i(x) = \frac{1}{1+(x-\mu)^T\Sigma^{-1}(x-\mu)}$, where μ and Σ are the mean and the covariance matrix of the part shape p^i. This allows to pass second-order information such as size and orientation of the part to the decoder. Note that this operation are fully differentiable.

4.3 Implementation Details

The image resolution is 128×128, but the resolution of corresponding part shapes is 64×64.

For the reconstruction loss $\mathcal{L}_{\text{rec}}$ we use the L_1 or L_2 distance. To prevent parts from trying to explain the whole image, instead of focusing on the object, we also restrict the reconstruction loss to an area around the part shape: a sum of Gaussian approximations around the means of the part shapes is folded with the loss.

In the decoder, the latent image Z is not only rescaled, but also filled with parts incrementally. At the lowest scale only some parts are inserted, with each scale parts are added until at the highest scale all parts are used. This makes the part decoding a hierarchical process. The underlying assumption is the parts exist at multiple scales. For landmark learning, we approximate the part shapes in the decoder in the bottleneck also with $\hat{p}^i(x) = \frac{1}{1+(x-\mu)^T\Sigma^{-1}(x-\mu)}$, but fix the covariance Σ to the identity matrix. Hence, effectively only information about the mean of each part shape can reach the decoder. This mean information is used as a landmark, so encouraging an accurate estimation of the mean through reconstruction is wanted.

To instantiate shape transformations π, one needs image pairs that show the same object in a different articulation or position: For static images an artificial thin-plate spline transform (TPS) can be applied, which generalizes rotation, scaling, translation. For video data adjacent frames exhibit natural shape transformations. The appearance transformation ϕ is encompassing a colour augmentation, contrast variations, and changes in brightness. In general, the more selective the transformation distinguishes shape and appearance, the more invariant the representation.

5 Experiments

5.1 Shape Learning

Fig. **??** visualizes the learned shape representation. To quantitatively evaluate the shape estimation, we measure how well groundtruth landmarks (only during testing) are predicted from it. The part means $\mu[\sigma_i(x)]$ (cf. (**??**)) serve as our landmark estimates and we measure the error when linearly regressing the human-annotated groundtruth landmarks from our estimates. For this, we follow the protocol of Thewlis *et al*. [**?**], fixing the network weights after training the model, extracting unsupervised landmarks and training a single linear layer without bias. The performance is quantified on a test set by the mean error and the percentage of correct landmarks (PCK). We extensively evaluate our model on a diverse set of datasets, each with specific challenges. An overview over the challenges implied by each dataset is given in Tab. **??**. On all datasets we outperform the state-of-the-art by a significant margin.

5.1.1 Landmark Discovery

On the object classes of human faces, cat faces, and birds (datasets CelebA, Cat Head, and CUB-200-2011) our model predicts landmarks consistently across different instances, cf. Fig. **??**. Tab. **??** compares against the state-of-the-art. Due to different breeds and species the Cat Head, CUB-200-2011 exhibit large variations between instances. Especially on these challenging datasets we outperform competing methods by a large margin. Fig. **??** also provides a direct visual comparison to [**?**] on CUB-200-2011. It becomes evident that our predicted landmarks track the object much more closely. In contrast, [**?**] have learned a slightly deformable, but still rather rigid grid. This is due to their separation constraint, which forces landmarks to be mutually distant. We do not need such a problematic bias in our approach, since the localized, part-based representation and reconstruction guides the shape learning and captures the object and its articulations more closely.

Human Faces

Human Bodies

Human, Olympic, Penn

Animal Faces/Bodies

Dogs, Cats, Birds

Composite Objects/Scenes

What is an object? What is a scene? compositional nature of reality Bird on twig object? Bird can also fly, but neural networks learn by correlation in data (-> ref to these ”failure modes” Dancing pair as object.

Object/Background Separation

Complexly cluttered background is actually favorable for the method. Correlations of object with background will belong to object.

Object Articulation

Object articulation makes consistent landmark discovery challenging. Fig. **??** shows that our model exhibits strong landmark consistency under articulation and covers the full human body meaningfully. Even fine-grained parts such as the arms are tracked across heavy body articulations, which are frequent in the Human3.6M and Penn Action datasets. Despite further complications such as viewpoint variations or blurred limbs our model can detect landmarks on Penn Action of similar quality as in the more constrained Human3.6M dataset. Additionally, complex background clutter as in BBC Pose and Penn Action, does not hinder finding the object. Experiments on the Dogs Run dataset underlines that even completely dissimilar dog breeds can be related via semantic parts. Tab. **??** and Tab. **??** summarize the quantitative evaluations: we outperform other unsupervised and semi-supervised methods by a large margin on both datasets. On Human3.6M, our approach achieves a large performance gain even compared to methods that utilize optical flow supervision. On BBC Pose, we outperform [**?**] by 6.1%, reducing the performance gap to supervised methods significantly.

5.1.2 Effect of Transformations

Parity

birds parity salsa parity

Rotation, Scaling, Translation

on Cats -> black cats different set of KP than rest -> connect these samples via transformation to reach intra-class consistency

Mimicking Appearance

Color, Contrast, Hue

5.1.3 Natural Changes

Video data: Penn, Own

5.2 Disentangling Generative Factors

Disentangled representations of object shape and appearance allow to alter both properties individually to synthesize new images. The ability to flexibly control the generator allows, for instance, to change the pose of a person or their clothing. In contrast to previous work [**?**, **?**, **?**, **?**, **?**, **?**], we achieve this ability without requiring supervision *and* using a flexible part-based model instead of a holistic representation. This allows to explicitly control the parts of an object that are to be altered. We quantitatively compare against *supervised* state-of-the-art disentangled synthesis of human figures. Also we qualitatively evaluate our model on unsupervised synthesis of still images, video-to-video translation, and local editing for appearance transfer.

t-SNE of Shape Representation t-SNE of Appearance Representation

5.2.1 Disentangling Pose and Appearance

On Deep Fashion [**?**, **?**], a benchmark dataset for supervised disentangling methods, the task is to separate person ID (appearance) from body pose (shape) and then synthesize new images for previously unseen persons from the test set in eight different poses. We randomly sample the target pose and appearance conditioning from the test set. Fig. **??** shows qualitative results. We quantitatively compare against supervised state-of-the-art disentangling [**?**] by evaluating *i)* invariance of appearance against variation in shape by the re-identification error and *ii)* invariance of shape against variation in appearance by the distance in pose between generated and pose target image.

ReID

t-SNE of IDs Own, Other (stronger statement) To evaluate appearance we fine-tune an ImageNet-pretrained [**?**] Inception-Net [**?**] with a re-identification (ReID) algorithm [**?**] via a triplet loss [**?**] to the Deep Fashion training set. On the generated images we evaluate the standard metrics for ReID, mean average precision (mAP) and rank-1, -5, and -10 accuracy in Tab. **??**. Although our approach is unsupervised it is competitive compared to the supervised VU-Net [**?**].

Pose

To evaluate shape, we extract keypoints using the pose estimator [**?**]. Tab. **??** reports the difference between generated and pose target in percentage of correct keypoints (PCK). As would be expected, VU-Net performs better, since it is trained with exactly the keypoints of [**?**]. Still our approach achieves an impressive PCK without supervision underlining the disentanglement of appearance and shape.

PCK Curve

5.2.2 Factorizing into Parts

Own Dataset: Move KP DeepFashion: exchange parts

5.3 Follow-Up

- make generative:(KP distribution estimation, variational features).
- make video generation possible (RNN on KP vector).
- better transformations -> appearance locally (around parts changed), appearance changed perceptually -> style transfer

6 Conclusion

-> need model-based approach (for counterfactual) make model as good as we can implementing as many assumptions as we can and only leave the rest to powerful model (humans also have brain structure and reasoning structure genetic)

need disentangling generative factors for imagination (i.e. synthesis) for manipulating factors mentally