UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

Multi-domain Inference and Generative Modeling for Hand Image Synthesis

by

Vassilios - Clitos Nicodemou

PhD Dissertation

Presented

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

Heraklion, January 2025

UNIVERSITY OF CRETE

DEPARTMENT OF COMPUTER SCIENCE

Multi-domain Inference and

Generative Modeling for

Hand Image Synthesis

PhD Dissertation Presented

by **Vassilios - Clitos Nicodemou**

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

APPROVED BY:

Author: Vassilios - Clitos Nicodemou

Supervisor: Antonis Argyros, Professor, University of Crete

Committee Member: Xenophon Zabulis, Research Director, FORTH

Committee Member: Georgios Tzimiropoulos, Associate Professor, Queen Mary University of London

Committee Member: Panos Trahanias, Professor, University of Crete

Committee Member: Nikos Komodakis, Assistant Professor, University of Crete

Committee Member: Grigorios Tsagkatakis, Assistant Professor, University of Crete

Committee Member: Anastasios Roussos, Research Director, FORTH

Department Chairman: Kostas Magoutis, Associate Professor, University of Crete

Heraklion, January 2025

Τα σημαντικότερα νευρωνικά δίκτυα που έχουμε να εκπαιδεύσουμε δεν είναι τεχνητά.

Acknowledgments

While the pursuit of this degree is inherently an individual endeavor, the journey towards its completion has been a collective effort. Along the way, I have been fortunate to receive guidance, encouragement, and support from many extraordinary individuals and institutions, without whom this thesis would not have been possible.

First of all, I would like to express my gratitude to my supervisor, Professor Antonis Argyros, who encouraged me and motivated me to become a member of the Computer Vision and Robotics Lab during my bachelor studies. He gave me the means, the right push, and the opportunity to become a member of the Computer Vision scientific community. I thank him for all the knowledge and expertise he conveyed to me throughout all my studies. Being an excellent professor and remarkable mentor, he played a decisive role in my choice to pursue this degree within this group.

I would also like to thank my advisor Iason Oikonomidis. His help, his support, and guidance played a decisive role in the completion of this thesis. I deeply appreciate the hours we spent studying and developing mathematical concepts, a challenging process for me, a relaxing activity for a mathematician like him. I am very grateful for his guidance and support over the years, not only in professional matters but also on a personal level.

I would also like to extend my appreciation to the members of my committee for accepting the invitation and devoting the necessary time and effort, positively contributing to the examination process. Specifically, I would like to thank the members of my advisory committee. Research Director Xenophon Zabulis for his useful comments and remarks. And Associate Professor Georgios Tzimiropoulos, for his valuable insights, which provided me with an important compass to guide the start of my PhD studies. Moreover, special thanks go to Research Director Anastasios Roussos, for his helpful comments and more importantly for giving me the opportunity to work with him and to be able to successfully extend part of my PhD work in other interesting areas. Also, I would like to thank Professor Panos Trahanias, Assistant Professor Nikos Komodakis, and Assistant Professor Grigorios Tsagkatakis for their contribution to the examination process.

Also, I would like to acknowledge the Institute of Computer Science of the Foundation for Research and Technology - Hellas (ICS-FORTH), the VMware University Research Fund (VMURF), and the Hellenic Foundation for Research and Innovation (HFRI) under the HFRI Ph.D. Fellowship grant (Fellowship Number: 803) for providing financial support and all the necessary equipment in order to accomplish this work.

A special thank you is also dedicated to all my friends for their support and encourage-

ment during all the years of my PhD studies. I would also like to thank all my colleagues and friends at the Human-Centered Computer Vision Lab for the pleasant and friendly environment that they created at work as well as for their valuable comments and insights during my PhD studies. Specifically, thanks to Nikolaos Kyriazis for his comments and his contribution to the conception of the idea of RV-modules. Paschalis Panteleris for his technical assistance and patience. Kostas Papoutsakis for his ongoing interest and talks over my work. Philippos Gouidis for adding philosophical depth to our conversations. And of course Ammar Qammaz, Georgios Karvounas, Kostas Bacharidis, Victoria Manousaki, and Panagiota Zotou for sharing with me the opportunity to present our works at international conferences, and providing insights, troubleshooting, and helpful conversations during my studies.

Last but not least, my heartfelt thanks go to my family. My parents, Vassilis and Natalia, for their continuous support, encouragement, and understanding and who were the first role models I aspired to follow. My beloved wife Anastasia, who supported me not only with her love, encouragement, and understanding but also with her brilliance as a mathematician. And of course, my son Andreas, his arrival at the end of my studies marked the beginning of a new chapter in my life, one filled with the most meaningful goals yet to come. With all my love, this dissertation is dedicated to all of you.

Abstract

The exponential growth of computer vision methodologies, particularly deep learning architectures, has significantly increased the demand for high-quality data across multiple domains, to serve as input or output modalities. While synthetic data generation has seen numerous developments in many areas, unfortunately the same does not apply to the field of human hand-related tasks. The few available synthesis approaches fail to achieve satisfactory realism and multi-domain variety. This dissertation aims to address this gap by proposing novel methodologies for hand image synthesis in the RGB and depth domains, focusing specifically on depth estimation, efficient probabilistic generative modeling, and controllable data generation.

To tackle the hand depth estimation problem, we aim at balancing accuracy and computational complexity. In that direction, we present a lightweight model based on the stacked-hourglass architecture for monocular RGB-to-depth estimation of hand images. Employing intermediate supervision and a staged learning approach, the proposed model achieves an accuracy of 22mm that other general-purpose depth estimating methods cannot achieve. To facilitate training and evaluation, we introduce HandRGBD, a new dataset of over 20,000 aligned hand image pairs of RGB and depth. With an accuracy comparable to that of low-cost depth cameras, this work bridges the gap between RGB and RGBD domains, enabling RGBD-based hand pose estimation methods to be applicable to RGB input.

To enhance generative methods with the imminent goal of enabling more accurate and controllable generation of images, we require direct manipulation of the latent representation of our data. Variational Autoencoders (VAEs) possess that property. For this reason, we augment this family of models by introducing Random Variable Variational Autoencoders (RV-VAEs), a novel formulation that treats artificial neural network activations as continuous Random Variables (RVs). We achieve this by integrating specially designed Artificial Neural Network modules, that are applied to RV operands using rules defined by the algebra of random variables. By optimizing over full distributions in the latent space rather than discrete samples, RV-VAEs incorporate mathematical priors, reducing computational load and improving reconstruction and generative performance.

Finally, we propose the Supervised Random Variable Variational Autoencoder (SRV-VAE), a novel framework for generating photorealistic hand images conditioned on precise pose annotations. The SRV-VAE manages to disentangle the pose and appearance of the input hand image, ensuring robust control over synthesis while maintaining realism

during inference. The model demonstrates efficacy in generating diverse hand images by utilizing the capabilities of RV-aware modalities. Moreover, using this framework, we are able to augment and diversify hand training sets with the purpose of enhancing pose estimation tasks.

Collectively, these contributions establish a foundation for advancing hand-related tasks in computer vision by offering solutions to multi-domain hand image synthesis. The presented methodologies bridge critical gaps in data generation and have the potential to meet the growing demand for high-quality hand data while addressing challenges in efficiency and realism, laying the foundation for future advancements in generative systems and hand analysis.

Keywords: Synthetic Data, Hand Depth Estimation, Probabilistic Generative Modeling, Random Variables, Conditional Generative Modeling, Hand Data Disentanglement, HandRGBD, VAE, RV-VAE, SRV-VAE

Supervisor: Antonis Argyros
Professor
Computer Science Department
University of Crete

Περίληψη

Η εκθετική ανάπτυξη των μεθοδολογιών υπολογιστικής όρασης, ιδιαίτερα των αρχιτεκτονικών βαθιάς μάθησης, έχει αυξήσει σημαντικά τη ζήτηση για δεδομένα υψηλής ποιότητας σε πολλούς τομείς, για την χρήση τους ως είσοδο ή έξοδο συστημάτων. Ενώ η παραγωγή συνθετικών δεδομένων έχει δει πολυάριθμες εξελίξεις σε πολλούς τομείς, δυστυχώς δεν ισχύει το ίδιο στον τομέα των εργασιών που σχετίζονται με το ανθρώπινο χέρι. Οι λίγες διαθέσιμες προσεγγίσεις σύνθεσης δεν επιτυγχάνουν ικανοποιητικό ρεαλισμό σε πολλαπλά πεδία (όπως το χρώμα, το βάθος, κλπ). Αυτή η διατριβή στοχεύει να αντιμετωπίσει αυτό το κενό, προτείνοντας νέες μεθοδολογίες για τη σύνθεση εικόνας χεριού στα πεδία του χρώματος και βάθους, εστιάζοντας συγκεκριμένα στην εκτίμηση βάθους, την αποτελεσματική πιθανολογική παραγωγική μοντελοποίηση και τη ελεγχόμενη παραγωγή δεδομένων.

Για να αντιμετωπίσουμε το πρόβλημα εκτίμησης βάθους χεριού, στοχεύουμε στην εξισορρόπηση της ακρίβειας και της υπολογιστικής πολυπλοκότητας. Σε αυτή την κατεύθυνση, παρουσιάζουμε ένα μοντέλο με μικρό πλήθος παραμέτρων εκμάθησης που βασίζεται στην αρχιτεκτονική στοιβαγμένης κλεψύδρας (stacked hourglass model) για εκτίμηση βάθους εικόνων χεριών από έγχρωμες εικόνες. Χρησιμοποιώντας ενδιάμεση επίβλεψη και μια προσέγγιση σταδιακής μάθησης, το προτεινόμενο μοντέλο επιτυγχάνει ακρίβεια 22mm που δεν μπορούν να επιτύχουν άλλες μέθοδοι εκτίμησης βάθους γενικής χρήσης. Για να διευκολύνουμε την εκπαίδευση και την αξιολόγηση του μοντέλου, παρουσιάζουμε το HandRGBD, ένα νέο σύνολο δεδομένων με περισσότερα από 20.000 ευθυγραμμισμένα ζεύγη έγχρωμων εικόνων χεριού και βάθους. Με ακρίβεια συγκρίσιμη με αυτή των χαμηλού κόστους καμερών βάθους, αυτή η προσέγγιση γεφυρώνει το χάσμα μεταξύ των πεδίων χρώματος και βάθους, επιτρέποντας τις μεθόδους εκτίμησης πόζας χεριού που βασίζονται σε βάθος να εφαρμόζονται στην είσοδο έγχρωμης εικόνας.

Για να βελτιώσουμε τις μεθόδους παραγωγής με επικείμενο στόχο να καταστεί δυνατή η πιο ακριβής και ελεγχόμενη παραγωγή εικόνων, χρειαζόμαστε άμεσο χειρισμό της λανθάνουσας αναπαράστασης των δεδομένων μας. Οι Αποκωδικοποιητές Διακύμανσης (Variational Autoencoders) διαθέτουν αυτήν την ιδιότητα. Για το λόγο αυτό, επαυξάνουμε αυτήν την οικογένεια μοντέλων εισάγοντας τους Αποκωδικοποιητές Διακύμανσης Τυχαίας Μεταβλητής (Random Variable-Variational Autoencoders (RV-VAE)), μια νέα εκδοχή που αντιμετωπίζει τις ενεργοποιήσεις ενός Τεχνητού Νευρωνικού δικτύου ως συνεχείς Τυχαίες Μεταβλητές (τ.μ.). Αυτό το επιτυγχάνουμε ενσωματώνοντας ειδικά σχεδιασμένες μονάδες Τεχνητών Νευρωνικών Δικτύων που εφαρμόζονται σε τ.μ. ως τελεστέους χρησιμοποιώντας κανόνες που ορίζονται από την άλγεβρα των τυχαίων μεταβλητών. Βελτιστοποιώντας τα μοντέλα μας στις πλήρεις κατανομές στον λανθάνοντα χώρο και όχι σε διακριτά δείγματα, τα RV-VAE ενσωματώνουν μαθηματικές προγε-

νέστερες κατανομές, μειώνοντας το υπολογιστικό φόρτο και βελτιώνοντας την ανακατασκευή και την απόδοση παραγωγής.

Τέλος, προτείνουμε τον Εποπτευόμενο Αποκωδικοποιητή Διακύμανσης Τυχαίας Μεταβλητής (Supervised Random Variable-Variational Autoencoder (SRV-VAE)), ένα νέο πλαίσιο για τη δημιουργία φωτορεαλιστικών εικόνων χεριών που εξαρτώνται από ακριβείς επισημειώσεις πόζας. Το SRV-VAE καταφέρνει να διαχωρίσει τη πόζα και την εμφάνιση της εικόνας του χεριού εισόδου, εξασφαλίζοντας ισχυρό έλεγχο κατά τη σύνθεση, διατηρώντας παράλληλα τον ρεαλισμό κατά την εκτίμηση. Το μοντέλο επιδεικνύει αποτελεσματικότητα στη δημιουργία διαφορετικών εικόνων χεριών χρησιμοποιώντας τις δυνατότητες των μοντέλων με επίγνωση στις τυχαίες μεταβλητές. Επιπλέον, χρησιμοποιώντας αυτό το πλαίσιο, είμαστε σε θέση να αυξήσουμε και να διαφοροποιήσουμε σύνολα δεδομένων εκπαίδευσης χεριών με σκοπό τη βελτίωση της εκτίμησης πόζας.

Συνολικά, αυτές οι συνεισφορές δημιουργούν ένα θεμέλιο για την προώθηση εργασιών που σχετίζονται με το χέρι στην υπολογιστική όραση, προσφέροντας λύσεις για τη σύνθεση εικόνας χεριού σε πολλαπλά πεδία. Οι παρουσιαζόμενες μεθοδολογίες γεφυρώνουν κρίσιμα κενά στη δημιουργία δεδομένων και έχουν τη δυνατότητα να ανταποκριθούν στην αυξανόμενη ζήτηση για υψηλής ποιότητας δεδομένα χεριών, ενώ αντιμετωπίζουν προκλήσεις σε σχέση με την αποτελεσματικότητα και τον ρεαλισμό, θέτοντας τα θεμέλια για μελλοντικές εξελίξεις σε συστήματα παραγωγής και ανάλυσης χεριών.

Λέξεις κλειδιά: Συνθετικά Δεδομένα, Εκτίμηση Βάθους Χεριού, Πιθανολογική Μοντελοποίηση, Τυχαίες Μεταβλητές, Μοντελοποίηση Υπό ΄Ορους, Διαχωρισμός Δεδομένων Χεριού, HandRGBD, VAE, RV-VAE, SRV-VAE

Επόπτης: Αντώνης Αργυρός
Καθηγητής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Computer vision is experiencing an era of ever-increasing innovation. In the course of the evolution of Artificial Neural Networks (ANNs) through Deep Neural Networks (DNNs), Large Language Models (LLMs), and Large Vision Models (LVMs) (Figs. 1.1 and 1.2), one fact becomes apparent. The necessity for data, whether it is for training or evaluation, is continuously increasing (Fig. 1.3). This is also emphasized by researchers that state the importance of data over methods [7, 53, 181]. With this exponential speed of growth, synthetic data can be practical for controlled and fast data acquisition. And since many models are deployed in more than one domain (not only in the Red Green Blue channel (RGB) domain), synthetic data is required to reside in multiple domains as well.

While many areas in computer vision have many methods of generating synthetic data, a more human-centered area, specifically the one that specializes in human hands, lacks this privilege. Numerous hand-related tasks rely on synthetic data, both those developed prior to and after the surge of ANNs.

Overall, it becomes clear that *multi-domain synthesis of human hand images* is relevant and necessary for the scientific community.

1.1 General Objective

While the need for data is addressed in its generality, either for 3D objects and models [22], autonomous driving models [37], learning optical flows [35], human full body [15], or even human faces [167], there is a shortage of works that focus on continuous generation of synthetic hand images. Moreover, datasets that have been created to facilitate synthetic hand images usually lack the desired realism [182], or are only in the RGB domain [87].

This dissertation aims to advance the field of computer vision and generative modeling that focuses on hands, by developing innovative methodologies that enable depth estimation, efficient learning, and controllable data generation for the task of hand image synthesis.

Specifically, for the depth estimation task for hands, we want to employ Convolutional

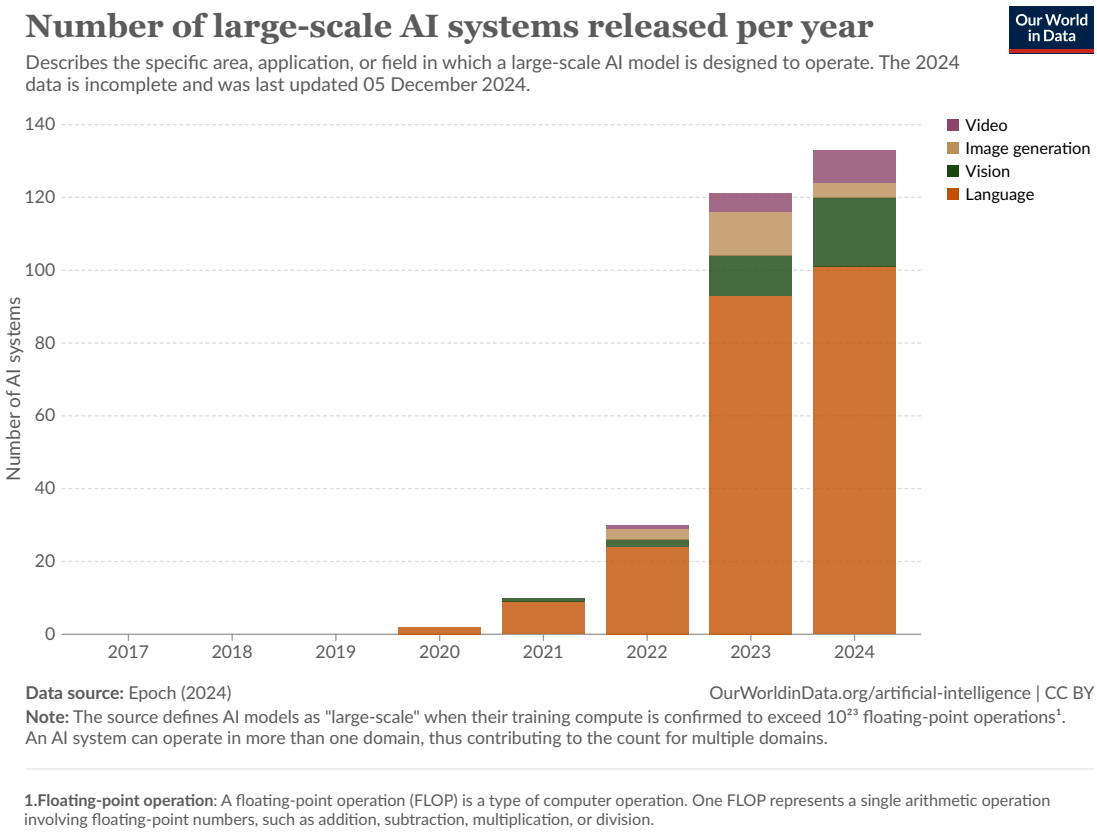


Figure 1.1: Vision-related Large-scale models correspond to an increasing proportion of the Large-scale AI models. Figure generated using: [50].

Neural Network (CNN) architecture and intermediate supervision techniques, to close the gap between RGB and Red Green Blue Depth channel (RGBD) modalities. The CNN architecture will ensure that our model will remain lightweight compared to the more demanding Transformer-based architectures. The intermediate supervision will be the main guidance for increasing accuracy. This new model must be trained and evaluated on appropriate annotated data that will help prove that hand depth estimation from RGB images is possible.

Developing synthetic data gives us further leverage, a control over the generated hand images. While this is the main advantage of synthetic data, the most productive way of doing it is not always straightforward. For hands, we are mainly interested in their pose (orientation, articulation) and their appearance (texture, shape, illumination, etc.). Having said that, in order to have control over those parameters, we require a method that manages to disentangle those attributes from an RGB image of a hand in an efficient way.

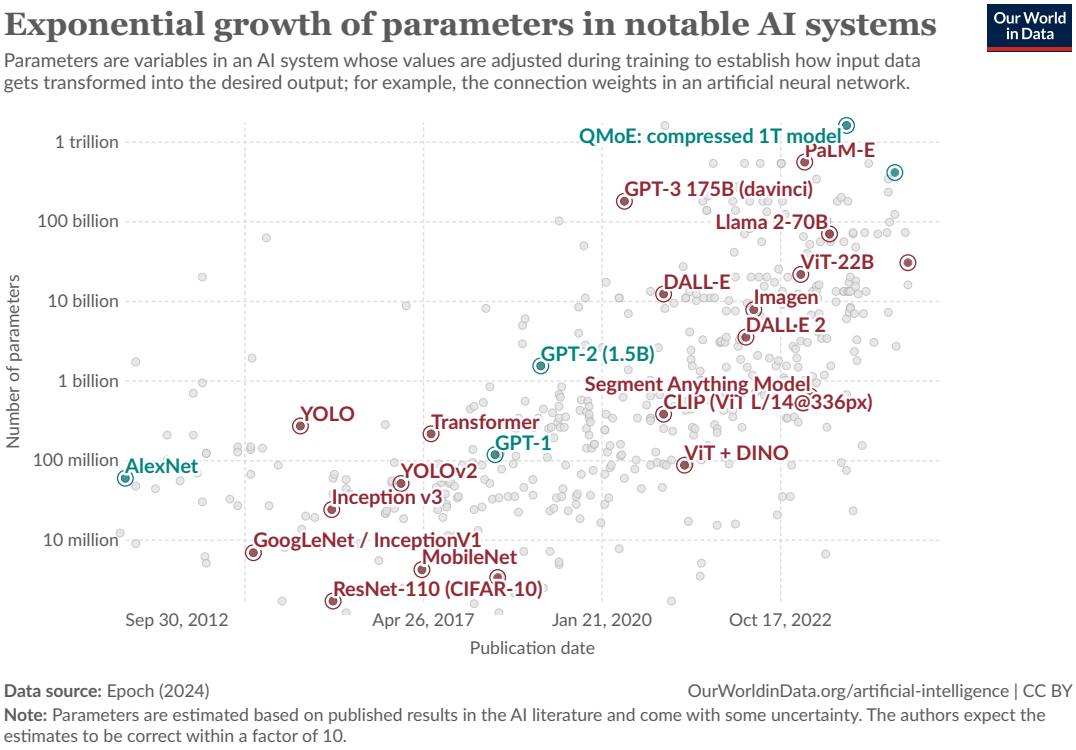


Figure 1.2: The growth of parameters in notable Computer Vision models (in red). Figure generated using: [50].

By inversing the problem we can find a function that renders back the original hand image of origin.

To achieve efficiency in the inverse, generative operation, the disentanglement should be accomplished in a deeper latent level. Models that have this latent configuration, and are capable of generative process are Variational Autoencoders (VAEs) [79]. However, the latent manipulation that we desire to achieve is restricted by the stochastic operations performed in the VAEs' architectures. To alleviate this stochasticity we must establish and use new deterministic formulations.

All the above tasks (depth estimation for hands, control of generated hand images, and efficiency of generative operation) should be addressed individually, by utilizing the benefits of ANNs. Using this kind of approaches and specifically of DNNs has its trade-offs. The deeper the networks the more parameters are needed, increasing computational cost and time. That is why we need to target them individually: a collective approach will result in explosive growth of parameters, diminishing efficiency, run-time, and even accuracy.

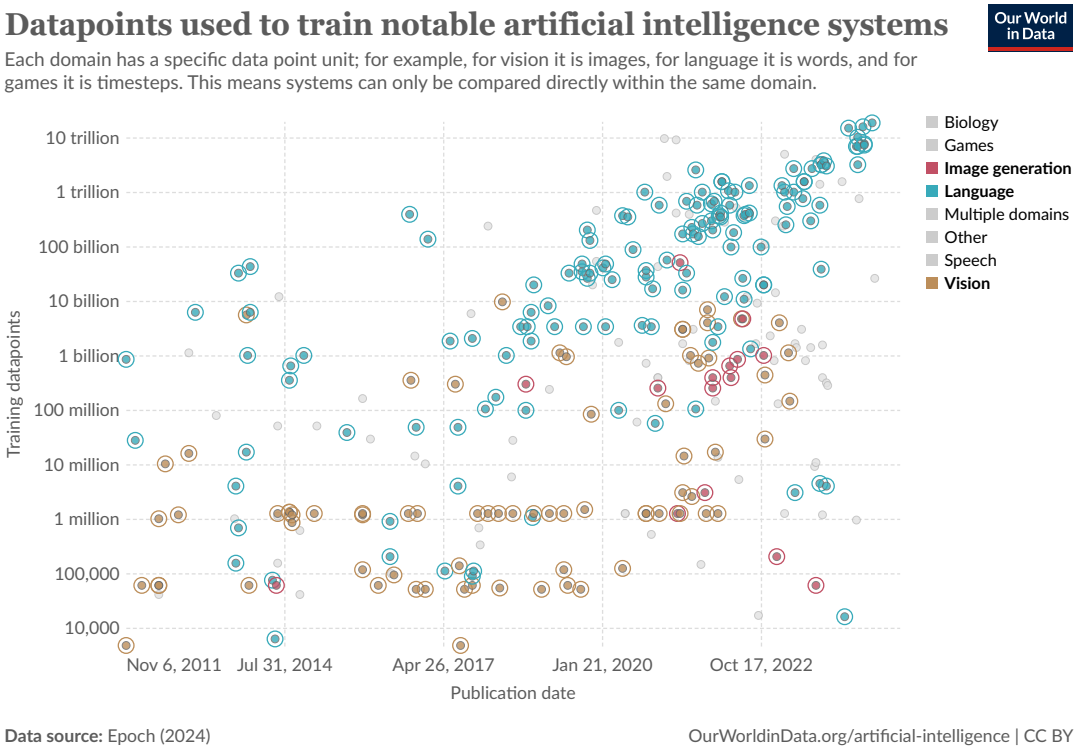


Figure 1.3: The increasing number of training sizes used for Computer Vision systems, compared to popular Language models. Figure generated using: [50].

1.2 Motivation and Vision

The main motivation for this thesis is to bridge the gap between the huge demand and lack of supply of hand image data. The importance of this task lies in the utility those data will provide to the research community that develops ANN models that require training and evaluation. Generating realistic images conditioned on precise annotations, such as hand pose, has practical uses in fields like Virtual Reality (VR)/Augmented Reality (AR), training simulations, and dataset augmentation. Such applications need to have control over the generated data in multiple domains.

For that reason, having data in more than one domain is very crucial, for methods that focus on utilizing multi-domain features. One of the most important feature domains other than RGB, is depth. This is attributed to the fact that many recent works depend on depth [17, 76, 119, 164, 170]. When we refer specifically to hands, depth information is also a necessary feature domain that is widely used in depth-related tasks and models [26, 27, 105–107, 126, 156, 177]. The most common way of acquiring depth is by measuring it with

depth-sensing systems. However this is not always practical, as it can become tedious, costly, and usually not feasible. For that matter, a method that goes directly from the RGB to the depth domain is more appealing.

Deep generative models, particularly VAEs, have revolutionized how we encode and reconstruct data. Yet, their reliance on stochastic sampling and limited optimization across full input distributions pose challenges. We aim to refine these models by integrating mathematical priors and treating encoded distributions as continuous random variables. This aligns with our broader vision of efficiently harnessing the latent space for manipulating the generative process.

By disentangling pose and appearance in latent space, we aim to provide tools that are not only accurate but also adaptable to novel conditions. This is intended to create synthetic data solutions that are indistinguishable from real-world counterparts while offering controlled diversity.

1.3 The Approach

The goal of multi-domain hand data synthesis is approached by three separate developed tasks. With each of them focusing on the objectives we mentioned earlier. While being separately implemented, they are thematically interconnected.

First, we present a direct approach targeted explicitly on human hands that infers depth from monocular RGB images. We achieve this with a lightweight CNN that employs a stacked hourglass model as its main building block. Intermediate supervision is used in several outputs of the proposed architecture in a staged approach. To aid the process of training and inference, hand segmentation masks are also estimated in such intermediate supervision steps, and used to guide the subsequent depth estimation process. In order to train and evaluate the proposed method we compile and make publicly available *HandRGBD*, a new dataset of 20,601 views of hands, each consisting of an RGB image and an aligned depth map. Based on HandRGBD, we explore variants of the proposed approach in an ablative study and determine the most accurate one. The results of an extensive experimental evaluation demonstrate that hand depth estimation from a single RGB frame can be achieved with an accuracy of 22mm, which is comparable to the accuracy achieved by contemporary low-cost depth cameras. Moreover, general-purpose depth estimation methods can not achieve such accuracies signifying the need for a direct and specialized approach. Such reconstruction of hands based on RGB information is valuable as a final result on its own right, but also as an input to several other hand analysis and perception algorithms that require depth input. In this context, the proposed approach bridges the gap between RGB and RGBD, by making all existing RGBD-based hand-related methods applicable to RGB input.

Among deep generative models, VAEs are a central approach in generating new sam-

ples from a learned, latent space while effectively reconstructing input data. The original formulation requires a stochastic sampling operation, implemented via the reparameterization trick, to approximate a posterior latent distribution. With our next work, we introduce a novel approach that leverages the full distributions of encoded input to optimize the model over the entire range of the data, instead of discrete samples. We treat the encoded distributions as continuous random variables and use operations defined by the algebra of random variables during decoding. This approach integrates an innate mathematical prior into the model, helping to improve data efficiency and reduce computational load. Due to the usage of random variables inside the proposed architecture, we name this family of new models as Random Variable Variational Autoencoders. Experimental results across different datasets and architectures confirm that this modification enhances VAE-based architectures' performance. Specifically, our approach improves the reconstruction error and generative capabilities of several VAE architectures, as measured by the Fréchet Inception Distance metric, while exhibiting similar or better training convergence behavior. Our method exemplifies the power of combining deep learning with inductive priors, promoting data efficiency and less reliance on brute-force learning.

Finally, we introduce a novel framework for generating photorealistic synthetic images of human hands conditioned to a precise pose annotation. We propose a Supervised Random Variable Variational Autoencoder, a model that disentangles and encodes the appearance and pose of the hand into separate components of the latent space. Appearance, representing individual subject traits, is unsupervised. Hand pose is strictly supervised and yields control over the synthesis process. Leveraging the robust Random Variable Variational Autoencoder variant, our architecture ensures stable training and accurate encoding of complex hand dynamics. Our model is capable of generating hand images of previously unseen hand poses for specific subjects. Experimental results indicate the model's efficacy in synthesizing realistic and varied hand images, holding significant promise for advancements in both academic research and practical applications such as data upsampling, where accurate hand pose and texture data is critical. We show that our method manages to aid the pose estimating task in some cases, by augmenting the training dataset. Moreover, we demonstrate the strengths that Random Variable Variational Autoencoders provide for facilitating such tasks.

All the proposed methods while implemented individually, contribute collectively to the problems that we address in this dissertation.

1.4 Contributions of this Dissertation

Our main contribution lies in the development of generative models for the problem of multi-domain hand image synthesis, offering both theoretical as well as application-based analysis of the topic. In more detail, the key contributions of this thesis are the following:

- The design and development of CNN architecture for the task of hand depth estimation from an RGB image.
- The compilation of *HandRGBD*, a publicly available hand dataset of 20,601 aligned RGB and depth images.
- The utilization of estimated depths by our model, on depth-based hand pose estimating methods.
- The ability to utilize the depth estimating model to other areas of use beyond hands as presented in the Appendix A.
- A theoretically justified approach for utilizing continuous distributions in ANNs and specifically in VAEs for incorporating inductive priors.
- The formulation and development of Random Variable Variational Autoencoders that achieve improved image reconstruction, generative results, and training convergence rate.
- The publication of the open-source framework of the Random Variable-aware modules, together with the presented Random Variable Variational Autoencoder modifications.
- The design and development of the Supervised Random Variable Variational Autoencoder family of conditional generative models that enable controlled and realistic hand image generation by disentangling hand pose and appearance in the latent space.
- The investigation and demonstration of the effectiveness of Supervised Random Variable Variational Autoencoders in generating high-quality hand images, even for previously unseen poses.
- The utilization of generated hand images for enhancing the performance of hand pose estimation models by data augmentation.
- The experimental demonstration of the benefits that the Random Variable-aware models provide for the task of conditioned hand image generation, compared to models that lack this Random Variable modification.
- The successful employment of our hand depth estimation method on the synthetically generated data from the Supervised Random Variable Variational Autoencoder model.

1.5 Outline of Dissertation

The rest of this dissertation is organized in the following way. Each of the three next chapters provides in detail the formulation of each individual problem we aim to tackle in this thesis, a brief literature overview for each case, the developed methodologies, and their experimental evaluations.

Specifically, Chapter 2 presents the task of estimating the depth information of hands. After being introduced to the literature overview, we proceed with the methodology of the proposed CNN model. We end this chapter with the experimental evaluation and use-case employment of our model.

Next, in Chapter 3 we present the problem of efficient generative modeling through our probabilistic approach. The introduction to related approaches is followed by the methodology of the proposed probabilistic modules and the formulation of the novel models. The chapter is finalized with the experimental evaluation and their results.

Chapter 4 reports and focuses on the last group of objectives stated by this thesis, those related to conditional probabilistic hand image generation. Again, after a brief introduction to the related literature, we present the proposed methodology, followed by an extensive experimental evaluation.

Finally, we conclude this dissertation in Chapter 5 with a brief synopsis of what we presented in this thesis along with the main contributions, and a basic future direction on the further extension of the proposed methods.

Chapter 2
Hand Depth Estimation

One of the fundamental tasks of computer vision is to acquire, process, analyze and understand digital images that are, essentially, projections of real-world scenes. Since they are projections, by definition some information is lost. In most cases, this information is the aspect of *depth*. The importance of depth can be seen when we are faced with the tasks of modeling, reconstructing, or understanding an observed scene. These tasks are interesting areas of research both from a theoretical and a practical point of view, having a large number of potential applications.

Another principal topic of great interest and importance in computer vision, is the study and understanding of human activities, since achieving such a high level of comprehension is one of the most difficult and important goals of the field. This can be achieved by monitoring and observing the state of the human body and body parts, either in 2D or in 3D. Particular emphasis is given to the human hands as the interpretation of their behavior is the key to understanding the interaction of humans with their environment through physical or digital context.

Until today, the best approach to acquire the depth of hands was by measuring it, using specific techniques that involve suitable hardware. Even though it was highly accurate, it was not always possible and efficient to do so. New techniques using Vision Transformer (ViT) models [36] are able to produce outstanding results [17, 76, 170] even for generic depth estimation. But even inferencing such models is resource-intensive. To the best of our knowledge, there are no methods that can estimate depth when provided with only a single color image, while also being lightweight.

Therefore, it becomes apparent that a direct lightweight method that specifically estimates depth information of hands from a single color image is essential.

There are many methods that try to estimate depth from color images [55, 67, 115, 134, 159,162]. Most methods depend on the target group of estimation. Meaning that, for different types of observed scenery we will have also a different approach. That is why a direct specific approach is preferable, as cross-target methods will have to be very hardware-demanding or may fail when targeted on deformable articulated objects, such as hands.

The vast number of research topics, publications as well as applications that make use of depth for hands [151], show the significant importance of solving the problem of depth acquisition for hands. Solving the aforementioned problem is interesting and useful not only for its practical use, but also for its contribution to research.

The majority of depth-based applications refer to modeling, reconstructing, proximity detection, gesture tracking and hand pose estimation mainly for Human-Computer interaction (HCI) [58,95,125,131,173,174]. Moreover, applications that build upon Virtual- and Augmented-Reality cases are in great need of depth to sense real 3D environments and reconstruct them in the virtual world [2,46,66]. Therefore, the depth of hands is also needed in order to include them in those environments for interactive applications.

Inferring depth information of hands from color information is a problem that presents important challenges. These challenges arise from the specific target group of human hands, but also from the task of depth estimation in general.

Human hands exhibit large differences in shape and appearance from person to person. They may vary in skin color, making it impossible to encapsulate all the differences into a single generic model. The shape/size of the hand plays a major role especially when we are referring to different age groups, as it is desirable to treat a child's hand and the hand of an adult uniformly, for example.

In addition, working with color images is also very challenging, as the same hand pose can be depicted in a large number of dissimilar images, due to different illuminations and various backgrounds. This makes the depth retrieval even harder for the same hand pose.

2.1 Literature Overview

Depth acquisition poses challenges that include a variety of forms in the field of computer vision. The first form to the best of our knowledge is that of shape from shading, with works in this area as early as 1970 [60, 61]. The inferred depth may be absolute [157] or relative [84], with the second case inferring the positions of points on a surface relative to some local coordinate system. Another approach inferred depth by interpreting line drawings or edge characteristics of a scene [10, 74]. These were the first techniques for depth acquisition, kick-starting the area of depth measurement and estimation.

Depth measurement can be achieved through depth sensing methods, that measure or sense depth often in real-time without any post-processing, usually with the help of special hardware [64, 69]. Depth estimation methods on the other hand use some form of visual input in order to estimate or evaluate depth, by processing the input based on a model or a learning method [33, 99, 133]. Though depth measurements provide the most accurate ground truth, they have some flaws that vary from method to method. Like the need of a controlled environment [69,88,171], or tedious setup [29,44,135], as we will elaborate. Depth estimation methods try to overcome these flaws of depth acquisition from

depth sensing, while trying to reach the estimation accuracy of measuring approaches.

While there might be numerous groups of depth estimating methods, such as estimation from Image Structure [45, 99], Defocus [77, 153], or Optical Flow [9, 109], we will focus mainly on methods that estimate depth from color.

Even though there are no direct methods that tackle the problem of depth estimation for hands, we should include methods that work with human hands and may somehow imply an indirect solution. There are methods that can be altered in order to derive some information about depth for hands [98, 110, 182], or might use depth indirectly for which we might have access [20, 47].

2.1.1 Depth estimation from color

There is more research breakthrough towards depth estimation rather than improving the existing depth measurement methods. This is the case also since the evolution of technology tends to create smaller and more lightweight innovations, meaning to achieve the same accurate results with the most simple and economical tools possible. Therefore, replacing complicated hardware with smart methods.

The desired task of depth estimation from color for hands can be derived from the more abstract target group of depth estimation for scenes to the more specific target group depth estimation for human body parts.

Targeted on scenes

Depth acquisition of scenes is the largest cluster of depth related methods because of its big demand. Many autonomous robotic systems rely on depth acquisition in order to move and avoid obstacles, outdoors and indoors. Especially lately with the huge increase in the development of autonomous vehicles, the need for fast and accurate depth acquisition is more necessary than ever.

The first line of works that stand out is employing Markov Random Fields (MRFs) at their core. In particular, Saxena *et al.* [132] used a supervised method of a trained MRF that exploits multi-scale local and global image features. By collecting relative and absolute features from image patches, they modeled depths at individual points as well as the relation between depths at neighboring points. Later an improvement of the same method was proposed in [133, 134] where the authors used this time superpixels, creating an assumption about the scene that it is made up of small planes. The latter method was used to create an application for making 3D scenes from 2D images and showed also how depth from stereovision can be improved.

The major outburst so far, of significant works and contributions began with the accession of Neural Networks and Deep Learning architectures. One of the first approaches for depth estimation that used a DNN architecture from a single image was by Eigen *et al.* [41].

This approach used a CNN architecture that follows a coarse to fine scheme. Although it was at an early stage of the DNN era, the results were very appealing and promising. The authors managed to improve their method [40], and enhanced it to solve more problems, such as normal estimation and context labeling. They achieved this by using as an initial step to their method, network models that achieve state-of-the-art performance in other problems from the literature, with the most successful being the Visual Geometry Group (VGG) network [140].

For the case of depth estimation, a significant number of a particular type of network architecture is observed to be used as a basic structure. The case is about Hourglass-like models. A name referring to the appearance of spatial resolution of their stacked layers as seen in [6, 90, 94, 104, 130]. The architecture consists basically of autoencoders that drop the spatial dimensions of the input (encode), and then raise them (decode) to the original dimensions in order to reconstruct the image based on the desired output. To that point, works like in [43] and [56] decode the desired depth map from an image, by inducing parameters into the decoded parts of the network. Specifically, in [43] they insert the camera model, and in [56] they use the focal length of an image to infer depth. Or as in [24] where the authors used a more sparse depth instead, as a prior knowledge inputted into the network to decode the final depth.

The use of geometric constraints proved to have notably significant results for Mahjourian *et al.* [92] in their method. The authors explicitly consider the inferred 3D geometry of the whole scene, and enforce consistency of the estimated 3D point clouds and ego-motion across consecutive frames of a video. Their suggested contribution lies in the proposed back-propagation algorithm for aligning 3D structures. They combine this 3D-based loss with 2D losses based on the photometric quality of frame reconstructions using estimated depth and ego-motion from adjacent frames. Therefore, the geometric losses are used as supervision to adjust unsupervised depth and ego-motion estimates by the neural network.

Further research by Casser *et al.* [21] made improvements with the use of geometric structure in the learning process. They achieved this by modeling the scene and the individual objects in it, alongside the objects' motions, ego-motion of the camera, and the depth of the scene in a principled way, from the video input. Furthermore, they refined their results and managed to adapt the learning on the fly to unknown domains, making this method robust for outdoor and indoor scenery. Altogether, their method managed to outperform supervised methods for depth estimation, giving excellent results for this task.

With the huge success of ViTs, new methods are developed that yield outstanding results even on generic depth estimation. Such are the works of Yand *et al.* [170] and Bochkovskii *et al.* [17] that perform generic depth estimation with state-of-the-art results. They managed to reach exceptional results mainly by increasing the number of parameters and training on a huge number of training samples. Therefore, the drawback of these

models is their large scale making them more difficult to deploy.

Targeted on human body parts

As we narrow down the targets of estimation towards our topic of depth estimation for hands, we should mention the research on the human body pose and shape estimation. New complexities and constraints are introduced now compared to the previous cases. Methods in this category have to deal with targets that express deformities, are articulated, dexterous in movement, and at most times self-occluded.

Shimshoni *et al.*in an early work [137] propose to reconstruct 3D bilaterally symmetric objects, such as faces. Their reconstruction algorithm combines geometric and photometric constraints appropriate for symmetric objects. The core idea behind their work is that an image of a bilaterally symmetric object can be regarded as two images of another object taken from two different viewpoints and two different illumination directions.

A work that emphasizes on estimating the face structure from a single color image is presented by Jackson *et al.* [67]. To do so, the authors use volumetric information to train a neural network that is based on a stacked hourglass architecture. They manage to regress the shape using a volumetric representation by guiding their network through training with 2D facial landmarks. This formulation improves their results in more difficult cases, like on facial expressions.

In parallel to the works on the human face, similar architectures are proposed targeting the human body such as [115]. The authors employ a model-based approach within a CNN architecture. The network learns the pose parameters and the shape parameters of a model and combines them into a single 3D mesh. By projecting the 3D model to the 2D input, and embedding this differentiable procedure into their network pipeline, a refinement step yields a significant improvement in the final results.

A more related method to the category of body parts, that tries to fit a volumetric shape estimation of a body is proposed by Varol *et al.* [163]. Their CNN-based architecture uses multiple types of information in order to fit the shape of a 3D body to the observations. Specifically, the network encapsulates the information of 2D pose, 2D body segmentation into body parts, 3D pose, volumetric representation, and re-projection from different views of the estimated 3D shape. All these embeddings, manage to yield excellent results for the given task of 3D volumetric body shape estimation.

The usage of ViTs was also used for depth estimation on humans. Where Khirodkar *et al.* [76] developed a family of models that target different estimations of human images (pose, depth, normals, and segmentation). For the depth, they trained a large ViT-based model on 300 million images of people. Despite the outstanding results, the large scale of the model (reaching 2 billion parameters) makes it again harder to deploy.

2.1.2 Depth estimation for hands

As described before, the problem of depth estimation for human hands from a single color image is very significant. This is supported not only by the number of possible applications, but also by the huge interest of the ongoing active research on the field in order to replace the old intrusive methods of depth acquisition. This means that the desired method is not hardware or setup dependent, and it only requires a single image of a hand as an input to directly estimate its depth, in order to be applicable to the real-world systems already mentioned. Alongside with the problems of depth estimation, hands exhibit an extra difficulty in finding a solution to this problem, due to their physical properties, as they are dexterous and therefore are subject to a lot of deformities and self-occlusions.

Directly approaching our task, we find methods that estimate depth of a hand from RGB that make use of a stereoscopic approach in their main core. Such a case can be seen in [11]. The authors use a Random Forest method to learn the disparity mapping between a pair of images, and with a targeted feature selection process they extract the best features for depth regression. Later, in [12], the authors improve their work by combining Conditional Random Fields (CRFs) and Regressive Random Forests. As to the authors, such an approach is not desirable due to the extra stereoscopic information needed that is not always reliable and possible to acquire.

There are many ways to derive depth of human hands from a single image indirectly. The most naive approach is to use already employed methods for tasks that target estimating the 3D pose and shape of human hands from color images. Provided the fact that this field sparks the interest of many researchers, there are many methods that deal with it. By having the estimated 3D articulations and joint locations, we can fit coarsely a model onto these points. From this known model, we can render an approximate depth image. Works like [98,110,182] specifically estimate only the 3D key-point/joint locations for single color images and can be used for estimating the depth of hands.

These methods though, do not take into consideration the shape and silhouette parameters of the hand. Without taking into account the hand's shape and appearance, we overlook an important property of the hand, that it is a deformable object. Therefore we should emphasize on works that consider these properties in their methodology, even if they do not focus on solving depth estimation directly.

One of these works is presented in [18]. The authors make use of a deep CNN-based framework that estimates 3D hand pose and shape from 2D color images. The final output of the system is the 3D mesh and skeleton of the hand based on a hand model.

Another work of great importance is presented by Cai *et al.* [20]. Their method also estimates 3D hand poses from 2D color images using a deep CNN-based architecture. Their key contribution is to use the hands' depth maps as weak supervision. Specifically, during training they employ a Depth Regularizer after deriving the 3D joints locations. Even

though during testing they do not use this part to output any depth.

An improvement of the previous work is presented by Ge *et al.* [47] where the authors estimate not only the 3D pose of the hand from 2D images, but also its shape. The proposed improvement uses a stacked hourglass architecture and embeds a 3D hand mesh into their loss that is derived by using 2D features from the input color image.

These methods are the closest in the relevant literature on roughly estimating the depth of hands from RGB images. Since they do not fulfill the requirements that we defined, we can not consider them as solutions. Nevertheless, their contribution and guidance for creating a direct approach are of utmost importance.

2.2 Hand Depth Estimation using Stacked-Hourglass Network

Solving the problem of hand depth estimation is both interesting and useful. When observing a scene using regular images, it is very appealing to be able to recover the suppressed depth information without stereo 3D reconstruction or structure from motion. Moreover, the recovery of this information may have significant impact on the solution of several practical problems. As an example, the hand depth information can be used to capture and understand hand movement within the 3D space, facilitating tasks such as 3D hand shape and pose estimation, hand-object interaction monitoring, etc, with immediate implications to areas such interaction [58], medical rehabilitation [39], computer games [5], and more. For several of the above applications, it suffices to have a 3D reconstruction of a hand, without necessarily solving the 3D hand pose estimation problem. For example, the 3D reconstruction of the hand suffices to support the realistic blending of a real hand with a virtual object (Fig. 3.5, third image) in AR/VR scenarios. 3D hand pose would be exploitable, but not required. Another interesting domain of support is the one of 3D hand pose estimation (Fig. 3.5, fourth image), where many applications depend notably on depth alone [23, 75, 107, 150]. At the same time, if the 3D reconstruction of a hand can be achieved from a single RGB frame, the inferred depth information can be fed to a depth-based 3D hand pose estimation method like the above. Therefore, a fast lightweight method that closes the gap between RGB and depth-based applications is essential.

In this work, we capitalize on recent advances in machine learning and propose a DNN architecture to tackle the problem of hand depth estimation from color images. We propose a method that accepts as input a conventional RGB image of a hand (Fig. 3.5, first image) and produces the depth map of the observed hand (Fig. 3.5, second image).

At the core of the proposed approach, a deep neural network undertakes the task of estimating the geometry of a hand observed in a single RGB image. A stacked hourglass model [101] is inspired from parts of the architecture in [67] and is used as our main building block. The resulting network accepts as input a regular RGB image and outputs the estimated hand depth map. The output of the network is a map of relative depths for all

Figure 2.1: Given an RGB image of a human hand our goal is to produce a depth map of the hand region. This can support AR/VR applications or 3D hand pose estimation. Note that applications such as AR/VR can exploit 3D hand pose estimation, but many can be supported using only hand depth information.

hand pixels of the input image. The absolute depth of the hand is a separate problem [157]. We tackle it through an extension of our method, that infers the absolute depth by learning the intrinsics of the input given only the bounding box of the hand (see Sec. 2.3.5). Intermediate supervision is used in several intermediate levels of the proposed architecture in a staged approach. To aid the process of training and inference, hand segmentation masks are also estimated in such an intermediate supervision step, and used as guidance for the subsequent depth estimation.

The training of such a method requires aligned RGB and depth information for a large number of hand views. Up to now, there is no publicly available dataset that is suitable for such a training process (i.e., requiring no significant preprocessing such as color-depth alignment, and having a large diversity of hand shapes, poses, illumination conditions, and backgrounds). This comes as a surprise, but can be explained by the fact that most of the hand-related works dealt with the problem of 3D hand pose estimation and tracking based on depth information. Therefore, annotation involves depth maps and the associated 3D hand poses and does not include color information. We, therefore, compiled *HandRGBD*, a dataset of $20,601$ pairs of hand color images aligned with their respective depth maps. We use *HandRGBD* to evaluate our approach in comparison with variants in the context of an ablation study. We show that hand depth estimation from a single RGB frame can be achieved with an accuracy of 22mm, which is comparable to the accuracy achieved by contemporary low-cost depth cameras. Moreover, we show that general-purpose depth estimating methods can not outperform our model even with $10\times$ more parameters.

Finally, we show that a 3D hand pose estimation method is only $2-7$mm more accurate when it estimates the 3D hand pose on real depth data, compared to when it is applied on the depth data estimated by our method. Thus, the proposed method is a significant step

towards turning an RGB camera into an RGBD one for hand analysis applications. Moreover, the proposed method makes all depth-based hand analysis methods exploitable on plain RGB input.

2.2.1 Ground Truth Annotation

The training data are assumed to be pairs of aligned RGB and depth hand images. The viewpoint of each image pair is considered to be identical, i.e., each RGB pixel essentially corresponds to the pixel at the same position in the depth map, as if the two streams were captured from the same center of projection. This kind of data can be obtained using common RGBD sensors like Microsoft Kinect2. Most commercially available RGBD cameras have different sensors for each modality, however the viewpoints are very close, and the availability of depth data enables the alignment of the two streams. To achieve this, an accurate intrinsic and extrinsic calibration of the two sensors is required. Given this, a reprojection of the depth map to the RGB image yields the correspondences between the two images.

Given this capturing process, the training data for a learning approach is already at a usable state and no further manual annotation is required. The only processing that is still required is the segmentation of the RGB and depth channels into foreground (hands) and background (non-hands), as well as the normalization of the depth range into relative depths. To facilitate this, it is assumed that the hand is the object closest to the camera, thus, foreground/background segmentation is easy to perform on the depth map. Specifically, the minimum value D_{min} in the depth map $D(i,j)$ is estimated, corresponding to the distance of the hands' point that is closest to the camera. The indices i and j run on the horizontal and vertical image dimensions. All pixels with depth value within a predefined threshold t to this minimum depth D_{min} are considered as the foreground H. The value $H(i,j)$ of the boolean foreground mask H at point (i,j) is defined as:

$$H(i,j) = D(i,j) < (D_{min} + t). \tag{2.1}$$

Working with depth maps in millimeters, it suffices to set $t = 300mm$, a maximum estimation of the possible depth difference within a hand. Since the RGB and depth images correspond pixel-to-pixel, the resulting binary segmentation H is valid for the RGB image, too.

Let us denote with $D[H]$ the depth map D masked with the foreground mask H. $D[H]$ is used to compute the average depth value $\overline{D[H]}$ of hand points. $\overline{D[H]}$ is then subtracted from D and a fixed scaling is applied to the depths, bringing the depth values in the range $[-1,1]$. Specifically, the relative depth map D_T that is used for training is

$$D_T(i,j) = c \cdot \left(D(i,j) - \overline{D[H]}\right), \tag{2.2}$$

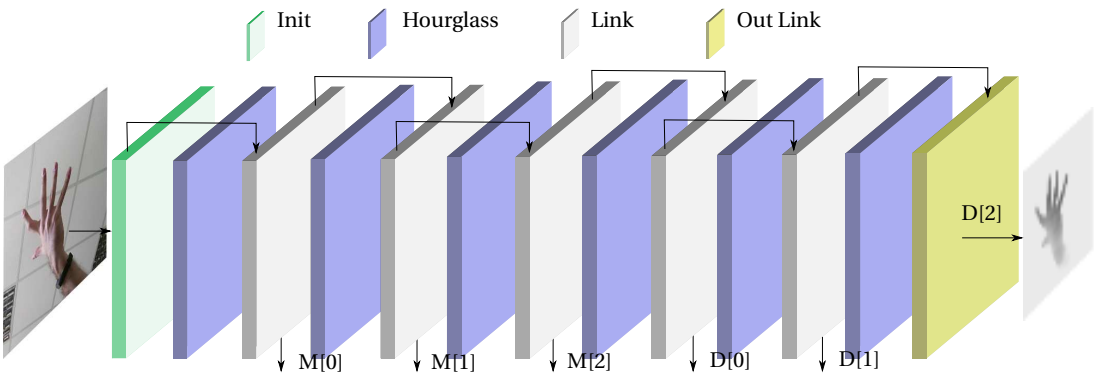


Figure 2.2: Stacked Hourglass Architecture: The proposed architecture with the intermediate supervision types. The input is preprocessed by some initialization layers ("Init", light green) that include a convolutional layer and two residual blocks (Fig. 2.3) and compute a feature map to be passed to the first hourglass (see Fig. 2.4) module. Its output is passed to layers that apply some additional convolution layers ("Link", gray) before passing it to the next hourglass module. The Link module also outputs a map to be intermediately guided. Skip connections are used parallel to each hourglass module. The first three outputs of the network target segmentation masks and the remaining three target depth maps with the latter being the final output of the network.

where c is a value that scales the depths in the range $[-1, 1]$. For all cases, it suffices to set c as the inverse of the maximum depth difference of an observed hand. When working with depth values in mm it suffices to use $c = 1/200$. Finally, the background pixels are set to 1, the largest value in the target range, which is essentially used to denote background areas.

2.2.2 Stacked Hourglass Architecture

The proposed network is based on the approach of stacked hourglass modules [19, 101]. Intermediate supervision is also applied to the output of each hourglass module, which is a commonly adopted strategy [101]. The resulting architecture is illustrated in Fig. 2.2. In the following description, the intermediate parts of the network are called stages.

The main building block of the proposed architecture is the hourglass network of [101] built using the residual block of [19][1]. Figures 2.3 and 2.4 illustrate the residual block and the network used. An hourglass module (Fig. 2.4) accepts as input a set of feature maps. The residual block of [19] proceeds by applying three successive sets of convolution, batch

[1]Implementation available online at `https://github.com/1adrianb/face-alignment`

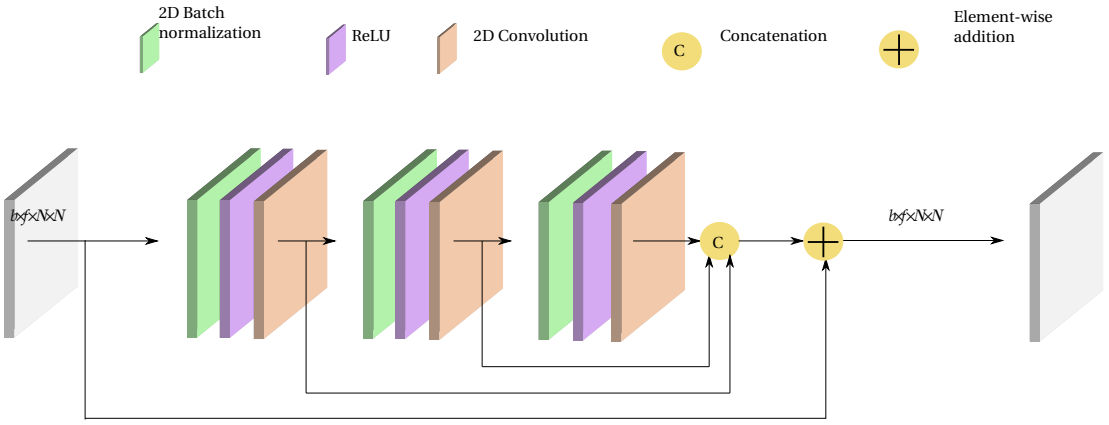


Figure 2.3: Residual block: The building block of the proposed neural network for hand depth estimation. The input is assumed to be a feature map of spatial dimension $N \times N$. In the figure, the feature count is f. The batch size b is also shown in the tensor dimensions. The output of the block is usually fed to more than one layer, e.g., to serve a skip connection, as shown here.

normalization and Rectified Linear Unit (ReLU) non-linearity operations, using also skip connections, similar to the DenseNet architecture [62]. This is shown in Fig. 2.3. After these operations, a down-sampling is performed, halving the input dimension. Parallel to this branch with a halved spatial dimension, a skip connection runs through another residual block. In total, four repeated residual blocks and resolution halving are applied, and four long-skip connections run in parallel, each at a different spatial resolution. After the last subsampling and application of a residual block, the reverse process is followed, doubling the spatial dimension by upsampling and applying new residual block operations. After each upsampling, the long-skip connection of the appropriate spatial dimension is added to the current feature map. After four upsampling operations, the original input spatial and feature dimension is again reached, forming the complete hourglass module.

For the proposed network, we stack 6 such hourglass modules, having a total of 6 stages of intermediate supervision. A convolution operation is applied to the input image to compute a feature map of the appropriate dimension to be the input of the first hourglass module. The reverse process is followed at the end of the network, and at the end of each hourglass module for intermediate supervision. Specifically, a single 1×1 convolution is applied, yielding a single-channel feature map. Each such output is trained against the foreground mask in the first stages, while the later stages are trained against the depth target. We set equal effort for estimating the mask and the depth, thus giving 3 stages for each such estimation.

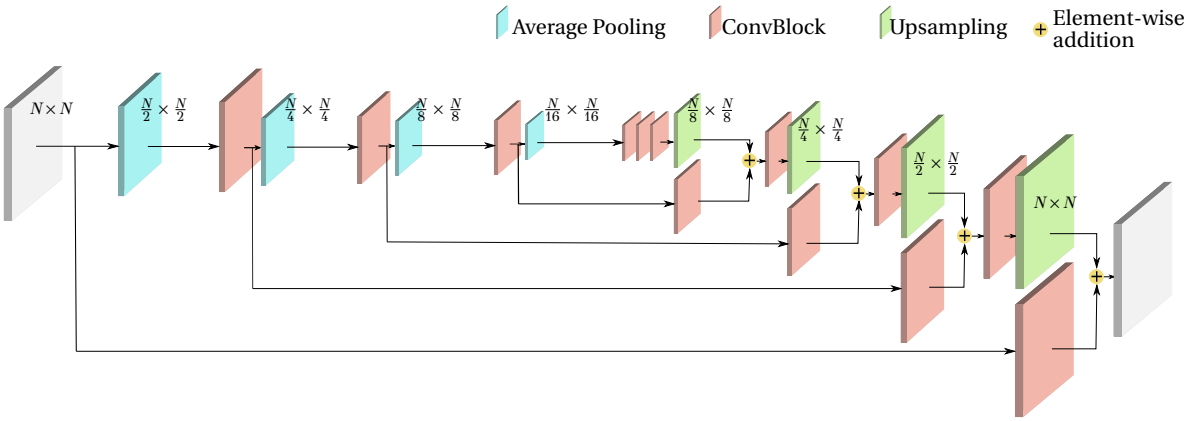


Figure 2.4: The hourglass building block that is used in the proposed network. Each input resolution shown in the figure is actually $b \times f \times sr$, where sr are the corresponding spatial dimensions illustrated. Its main building block is the residual block, illustrated in detail in Fig. 2.3. The main idea is to successively lower the spatial input resolution for a total of four input-halving steps. After this, the reverse procedure is followed to reach again the input resolution.

2.2.3 Loss Function

Each stage has its own target output, and thus, its own loss. The global loss function of the network is the sum of the individual losses. For each stage, regardless of the type of intermediate supervision, its loss is obtained by comparing the two images, the predicted and the target image. We define the loss of each stage as the Mean Squared Error (MSE) of the target and the output. The final loss function L is formulated as such:

$$L(m, d, \tilde{m}, \tilde{d}) = \sum_{k=1}^{S_D} \frac{(\tilde{d}_k - d)^2}{||N||} + \sum_{l=1}^{S_M} \frac{(\tilde{m}_l - m)^2}{||N||}, \tag{2.3}$$

where d and m are the target depth and mask, each having N pixels, S_D and S_M are the total number of depth and mask stages and $\tilde{d}_k$, $\tilde{m}_l$ are the estimated depths and masks for the kth and lth stage for $k = 1 \ldots S_D$ and $l = 1 \ldots S_M$.

2.2.4 Data Augmentation

During training, data augmentation aims to enrich the diversity of the training set and increase the generalization capability of the trained network. In our case, the input is

regular RGB images, and common augmentation practices apply. Specifically, we apply:

(a) random horizontal flip (so that we don't have to capture both hands from a subject)

(b) random rotation

(c) random crop

(d) random color jittering (to capture the widest possible range of skin tones and different illumination cases).

The geometric transformations are applied to both the RGB and the depth maps, ensuring pixel-to-pixel correspondence. Finally, all data are resized to fit the network's input and output dimensions.

2.3 Experimental Evaluation

In some first experiments, we attempted to reconstruct the depth of a hand based on general-purpose depth estimation techniques such as the works of [86, 103]. Specifically, when based on their pre-trained models, the above methods yield an error E of 47.59mm and 35.24mm, respectively, on the test set of *HandRGBD* dataset. We show that on the same dataset, the proposed method reduces this error by almost 50%. Another category of experiments on the *HandRGBD* dataset assess the adopted design choices and define meta-parameters of the proposed approach in an ablation study. We also evaluate our approach on the publicly available Stereo Hand Pose Benchmark (STB) dataset [177]. Finally, we assess the potential of the proposed method to support other methods that perform depth-based 3D hand pose estimation.

2.3.1 Hand-related Datasets

Works related to hand appearance and shape modeling as well as pose estimation, require datasets appropriately annotated with ground truth for the purposes of objective, quantitative comparison of competitive approaches, and also - whenever applicable - for training. Therefore, numerous datasets have been proposed so far in the relevant literature [8, 38, 51, 71, 127, 139, 146, 149, 152, 154, 156, 160, 168, 175–177, 182]. Input modalities such as monocular RGB, stereo, multiview, and depth are covered. Also, scenarios including egocentric viewpoint, hand-object interaction, and hand-hand interaction are available.

The training and evaluation tasks of the problem we are addressing in this work call for a dataset that includes aligned RGB and depth observations of hands. The RGB input should be unaltered, since the goal is to apply our method to regular color input. Some datasets [154, 156] warp the RGB image to the depth map, introducing big black holes in

Table 2.1: Datasets on human hands. For the purposes of this work, aligned pairs of RGB and depth data are required.

Dataset	RGB	Depth	Alignment
Gomez [51]	✓	-	-
Simon [139]	✓	-	-
Bambach [8]	✓	-	-
Dreuw [38]	✓	-	-
Yuen [176]	✓	-	-
Tang [152]	-	✓	-
Sun [149]	-	✓	-
Yuan [175]	-	✓	-
Xu [168]	-	✓	-
Tompson [156]	Warped	✓	-
Tkatch [154]	Warped	✓	-
Rogez [127]	✓	✓	-
Sridhar [146]	✓	✓	-
Zhang [177]	✓	✓	-
Kanhangad [71]	No BG	✓	✓
Zimmermann [182]	✓	✓	✓
Tzionas [160]	✓	✓	✓

the images that defeat this goal. Another dataset [71] segments the hand in the image and masks the background with a black color. Given that one of our goals is also to learn this segmentation, the dataset becomes unusable. Two additional requirements are the presence of multiple actors and close-up views of the depicted hand(s), so that details on the variation of hand shapes across humans and under articulation are adequately captured.

Table 2.1 presents the most relevant datasets to our work. Columns list some of the requirements mentioned above (availability of RGB and depth data and their alignment). Evidently, only the datasets by Zimmerman and Brox [182], called Rendered Handpose Dataset (RHD) and Tzionas *et al.* [160], called "Hands in Action" have aligned RGB and depth data. Unfortunately, the RHD dataset [182] is synthetic, and, although it has a large variation in hand sizes, shapes and appearances, it is of rather low resolution (320×240) and contains distant views of a hand. The Hands in Action dataset [160] contains real world data, and the depth is captured by a structured light sensor. The actor diversity is small and the view is not closeup, in images of resolution 640×480.

The *HandRGBD* Dataset: Despite the existence of several hand datasets, none of them fully covers the requirements of this work. Consequently, we resorted to creating *HandRGBD*, our own dataset of aligned RGB and hand depth images.

Figure 2.5: Samples of RGB images from our *HandRGBD* dataset.

As the capturing device, we employed a Kinect V2 [1] because of its high quality color camera, and the Time of Flight (ToF) depth sensor. Among the available options, this provided the best combination of image and depth resolution and quality. The native Software Development Kit (SDK) does not offer an alignment of the depth data to the RGB image, only the opposite, resulting in black holes in the RGB image. Therefore, we used the library libfreenect2 [42][2] that supports this functionality, simultaneously scaling and aligning the depth on the color image.

HandRGBD consists of 20, 601 images along with their depth maps. These come from 47 sequences, each consisting of approximately 450 frames. In total, 17 subjects (13 male, 4 female) contributed to the dataset. The depicted hands are in closeup view, in distances ranging from $40cm$ to $100cm$ from the sensor. Some of the captured images contain two hands that interact (strongly, in some cases). All subjects were recorded more than once and in a variety of illumination conditions. Special care was taken to capture the hands in front of different background scenes, facilitating the generalization of foreground/background segmentation. The subjects were instructed to keep their hand(s) roughly in the center of the camera field of view, but some images were also captured with hands close to the image edges. The subjects were also instructed to perform free hand motions, exploring as much as possible the hand articulation space. Some indicative samples of the dataset are illustrated in Fig. 2.5.

The STB Dataset: Given that the STB dataset by Zhang *et al.* [177] is annotated with 3D hand poses, we employed it, mainly for the quantitative evaluation of the proposed approach on the task of supporting 3D hand pose estimation. Since this dataset does not have aligned RGB and depth streams, we manually mapped the depth map on the pixels

[2]Available online at `https://github.com/OpenKinect/libfreenect2`.

of the RGB images using the provided calibration.

2.3.2 Training Details

We implemented the proposed approach using the PyTorch framework [112]. The Adam optimizer was used to train it for 100 epochs, with a learning rate value of 10^{-3}, weight decay of 10^{-5} and a learning rate scheduler with $\gamma = 0.5$ applied every 30 epochs. For training, we employed an Nvidia GTX 1080 Ti GPU. On that machine, each epoch took about 825 seconds. For all the experiments, the input size to the network was a 256×256 RGB image, and the output was a 64×64 depth map. For these resolutions, the inference time for a single image ranges from 7 to 31ms, depending on the number of stages (see also Tab. 2.2).

We split *HandRGBD* into a training set of $19,104$ samples and a test set of $1,497$ samples, from sequences that are not included in the training set. Specifically, we left aside 3 sequences of our dataset for testing, with 1 female and 2 male subjects. This choice was made because of the ratio between the recorded sequences of female and male subjects. Moreover, each of the three test sequences have backgrounds that appear only in these three sequences. Following the same reasoning, for the experiments on the STB dataset, 11 out of 12 sequences ($16,500$ samples) were used for training and the remaining sequence ($1,500$ samples) was used for testing.

As already mentioned, data augmentation was used in order to increase the generalization potential of the network. Each training sample was randomly flipped horizontally with a probability of 0.5. Also, a random rotation in the range of $[-90°, 90°]$ was applied. For the random cropping, a bounding box of size 0.8 of the original size was selected. Finally, a random intensity value in the range of $[-20, 20]$ for each color channel was added for color jittering.

2.3.3 Evaluation Metrics

Depth estimation accuracy: For each hand pixel we consider the absolute difference between ground truth and estimated depth. The first error metric E (in mm) is the average of all these differences for all *actual hand pixels* and all frames of a test set. A second error metric is the percentage $F(e)$ of hand pixels in the test set for which the absolute difference between ground truth and estimated depth is less than a threshold e.

Hand/background segmentation: The proposed method also produces a segmentation of the hand regions from the background. To assess this, we compute the *Intersection over Union (IoU)* criterion for this classification.

Table 2.2: Ablative study for the proposed hand depth estimation.

Architecture	**Error E (mm)**	***IoU***	**Runtime**
0 Mask, 1 Depth Stage	39.75	0.62	$7.07ms$
1 Mask, 2 Depth Stages	33.16	0.65	$16.35ms$
1 Mask, 3 Depth Stages	29.04	0.70	$20.82ms$
1 Mask, 4 Depth Stages	28.05	0.73	$25.73ms$
1 Mask, 5 Depth Stages	28.83	0.72	$30.93ms$
2 Mask, 4 Depth Stages	25.00	0.73	$31.21ms$
3 Mask, 3 Depth Stages	**24.64**	**0.81**	$31.01ms$
4 Mask, 2 Depth Stages	34.42	0.68	$31.51ms$

2.3.4 Ablative Study

We evaluate different architectural choices (Sec. 2.2.2) based on a subset of *HandRGBD*. Specifically, variants of the proposed method were trained on $4,500$ images (a subset of the main, training dataset) and tested on 500 separate images (one of the dataset's test sequences).

An important hyper-parameter of the proposed network is the number of intermediate supervision stages that target the mask segmentation. Experimenting with different training strategies for the proposed network, it became apparent that the hand segmentation mask is an important cue for the task at hand. In a preliminary experiment, the ground truth segmentation mask was provided as a fourth channel concatenated along with the RGB image to the network. This experiment lowered significantly the depth estimation error, indicating that the segmentation mask is indeed useful. It is therefore important to use this cue as an intermediate supervision target, since it aids the task of the network.

In a network with a fixed number of hourglass modules, some of the first hourglass module outputs target segmentation masks, and the rest target depths. We performed an experiment to determine the optimal number of stages for each of the two tasks, experimenting also with the total number of hourglass modules. The results of these experiments are summarized in Tab. 2.2. The results of the highest accuracy are highlighted in bold font. From this information we can conclude that, in fact, the segmentation cue is equally important to the depth map itself. The best performing network with six stages was trained with the first three stages targeted as segmentation masks and the rest targeting depth.

2.3.5 Hand Depth Estimation Accuracy

Relative depth estimation accuracy: We explored the performance of the best performing variant (line 7 in Tab. 2.2) when trained on the full training set of *HandRGBD* (Sec. 2.3.2). The depth estimation error was $E = 22.88$mm and the estimated *IoU* was equal to 0.84.

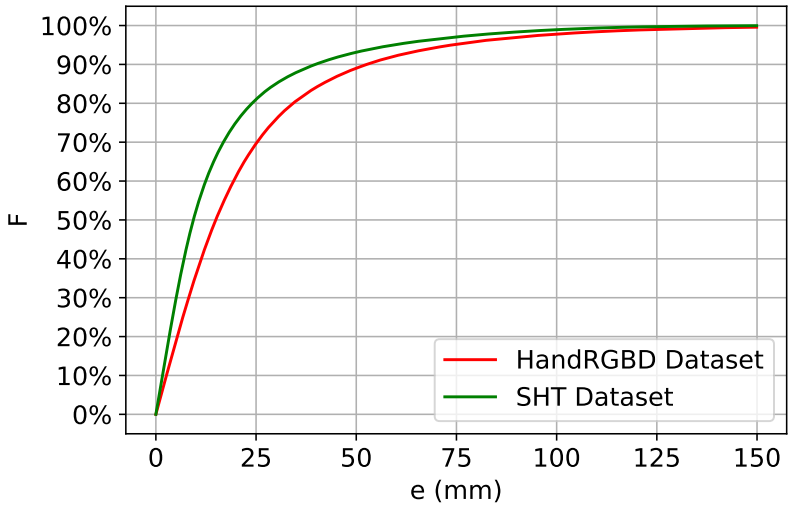


Figure 2.6: The error metric $F(e)$ for the depth accuracy estimation experiment on the *HandRGBD* and STB test sets.

When the same variant is trained on the training set of STB and tested on the corresponding test set, we obtain $E = 16.40$mm and $IoU = 0.88$. Figure 2.6 shows the metric $F(e)$ for both experiments. Focusing on the cross-dataset generalization between ours, *HandRGBD*, and STB, when trained on the former and tested on the latter, our model performed with error $E = 24.63$ and $IoU = 0.74$. Moreover, we trained our method using the training sequences of both datasets and testing on their respective test sequences, resulting in a depth estimation error E of 19.14 and IoU equal to 0.87.

Absolute depth estimation accuracy: A slight modification of the proposed network enables the estimation of an absolute depth map of a hand. The last feature vector of the proposed architecture is used to estimate (a) the relative depth map using a 1×1 convolution as already described, and (b) a single absolute depth value, using as additional input the bounding box of the observed hand within the input frame. Specifically, three stages of convolution and max-pooling are applied, and then two fully connected layers are used to estimate a single value, the median depth, using the quantity $\overline{D[H]}$ of Eq. (2.2) as ground truth. In total, the additional parameters required for this branch are less than 5×10^5, a small percentage of the 3.55×10^7 parameters for the proposed setup using 3 mask and 3 depth stages. Using this modification to the proposed architecture, we achieve an average absolute depth error E of 28.27mm on the test set of STB with IoU equal to 0.86.

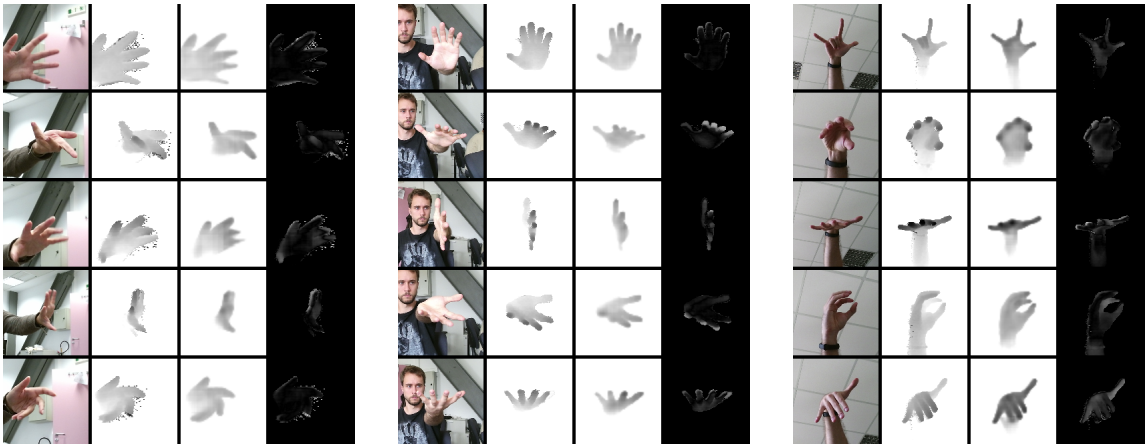


Figure 2.7: Indicative depth estimation results on the 3 test sequences of *HandRGBD*. For each sequence (4 columns per sequence) we show the RGB input (1st column), ground truth depth (2nd column), estimated depth (3rd column), and the difference between ground truth and estimated depth (4th column).

Table 2.3: Comparison of our stacked hourglass model, with other pre-trained general-purposed depth estimating methods on the STB [177] dataset.

Method	**Error E (mm)**
Niklaus *et al.* [103]	47.59
Lee *et al.* [86]	35.24
Ours	**24.63**

2.3.6 Comparison with General-purpose Depth Estimating Methods

We investigated the employment of other methods focused on general depth estimation, on hand depth estimation. Specifically, we used the pre-trained models by Niklaus *et al.* [103] and Lee *et al.* [86] and tested them on the test set of the STB dataset [177]. We also trained our model on the HandRGBD dataset and employed it on the test set of STB dataset, as well.

In Tab. 2.3 we report the average depth differences. As we can observe, the general-purposed methods can not outperform our model in accuracy. Moreover, our model has $10\times$ less parameters than the other competing methods.

2.3.7 Supporting 3D Hand Pose Estimation

We assessed the quality of the depth estimated by the proposed method by evaluating the extent to which it can support depth-based 3D hand pose estimation. To do so, we em-

Table 2.4: 3D hand pose estimation error (in mm) of Pose-REN [23] on the STB dataset for relative and absolute depths. **C1**: ground truth depths, **C2**: depths estimated by our method.

	Relative depth		**Absolute depth**	
	C1	**C2**	**C1**	**C2**
3D hand pose error	47.87	49.70	51.67	58.26

ployed the test set part of the STB dataset on which we applied the Pose-REN 3D hand pose estimation method of Chen *et al.* [23]. We chose this method because it is a recent approach that achieves close to state-of-the-art hand pose estimation and uses depth information. We applied Pose-REN in two different experimental conditions: **C1**, on the actual depth information of the test set as this was measured by the Intel Real Sense F200 sensor, and **C2**, on the depth that has been estimated by our method. By doing so, we can assess the potential of our method to provide depth maps that are usable by higher-level hand perception methods. We quantified the 3D hand pose estimation error by measuring the average distance of the estimated hand joints from their ground truth locations. We did that for the case of (a) relative depth estimation and (b) absolute depth estimation. For (a), the estimated hand is assumed to be located at its ground truth position. For (b), the accuracy of the estimated 3D hand pose is affected by errors in the estimation of the absolute depth.

Table 2.4 summarizes the obtained results. As expected, 3D hand pose estimation is more accurate in the case of relative depths compared to the case of absolute depths. However, the discrepancy between the conditions **C1** (ground truth depth) and **C2** (depth estimated by our method) is smaller. Thus, Pose-REN is only $2-7$mm more accurate when applied to real depth data, compared to when it is applied to the depth data estimated by our method. This is a strong indication that the proposed method can support 3D hand pose estimation.

2.3.8 Qualitative Results

Figure 2.7 shows representative depth estimation results on three sequences of the test set of *HandRGBD*. For each sequence, we show the input RGB image, the ground truth depth map, the estimated one, and their color-coded difference. It can be verified that the depth maps estimated by our method are very close to the ones measured by the depth sensor.

In addition, Fig. 2.8 shows some representative depth estimation results on the STB dataset. For each RGB input we present the ground truth depth map, the estimated depth map, and their color-coded difference. Furthermore, the two last columns show the result of applying the Pose-REN method to the ground truth and under condition **C1**, respec-

tively. Again, the estimated depth map and the sensor-captured depths are very similar. Moreover, it can be noted from the absolute differences figure, that parts of the sleeves were considered as ground truth parts of the hand, hurting the depth estimation errors, while not being technically part of this work's scope. Nevertheless, we can observe that despite their slight difference, a hand pose estimation method still manages to have appealing results on the estimated depths as well.

More experimental results, including representative depth estimation results on the STB dataset together with their corresponding hand pose estimations, are available in the supplementary video[3].

2.4 Summary

In this chapter, we presented a method that has been specifically designed to estimate the depth map of a human hand based on a single RGB frame. The proposed method consists of a specially designed convolutional neural network that has been trained and evaluated on *HandRGBD*, a new dataset of aligned RGB and depth hand images. Extensive experiments evaluated design choices of the proposed method, verified its depth estimation accuracy and provided evidence on the potential of the method to support existing depth-based hand pose estimation methods. The obtained results demonstrate that for the specific context of hand observation, the proposed method constitutes an important step toward turning a conventional RGB camera into an RGBD one using lightweight and accessible architecture. Furthermore, the experimental evaluation of the proposed approach shows that the task-specific intermediate supervision using the hand mask visual cue is more beneficial for the training process than directly using the target depth map for intermediate training.

The work in this chapter represents a key step in a larger approach toward advanced hand image synthesis. By bridging the gap between RGB and depth modalities, we are moving closer to a comprehensive, multi-domain synthesis pipeline. Completing this step enables us to shift our focus more directly to generative modeling within the RGB domain, which is essential for expanding the breadth and quality of hand data across multiple imaging modalities. Consequently, the following chapter will delve into this progression, exploring generative techniques in the RGB domain by leveraging probabilistic modeling.

[3] `https://youtu.be/q0sw8dZ3LlU`

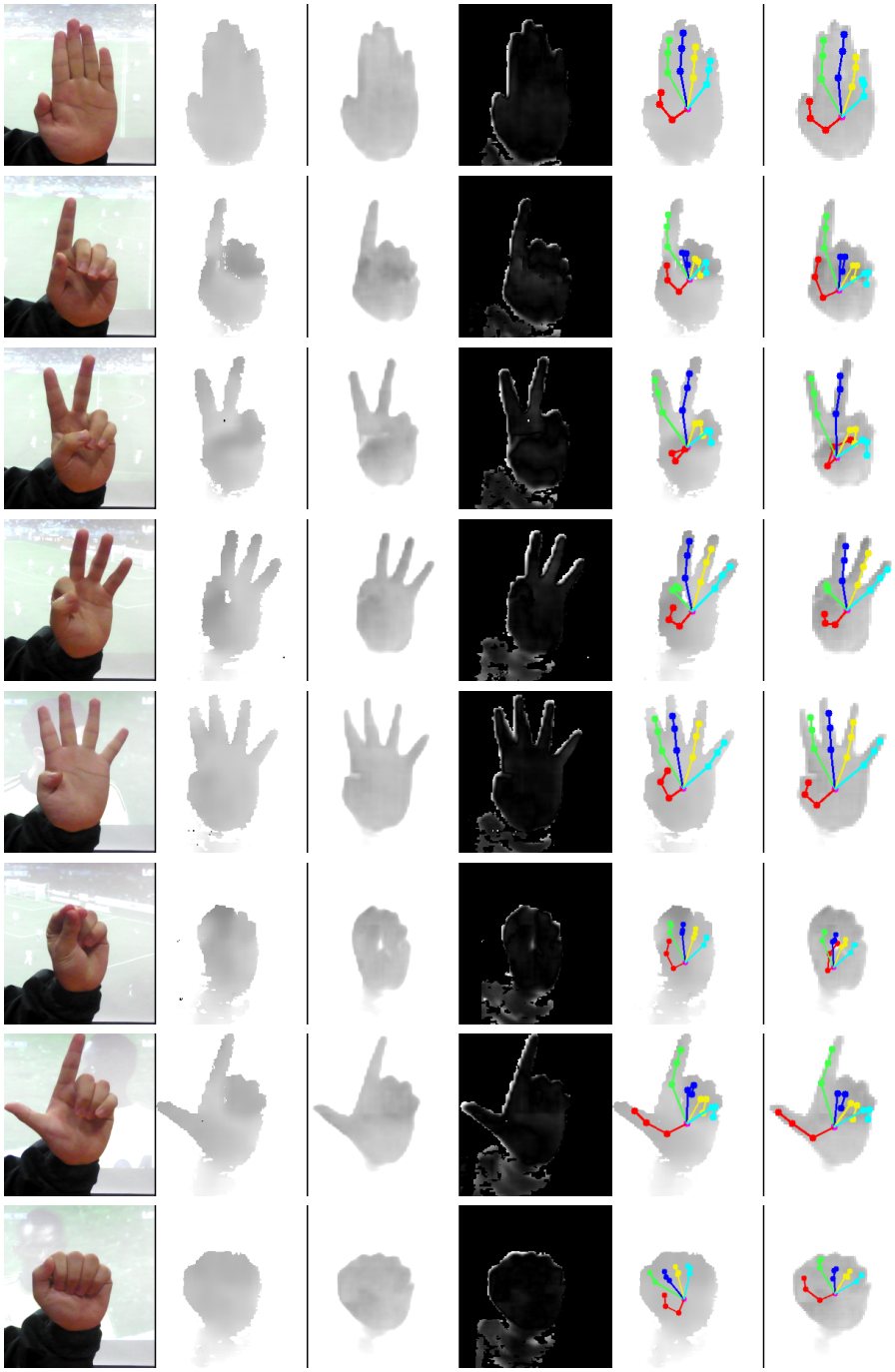


Figure 2.8: Indicative depth estimation results on the STB testset. For several different frames (rows), each column from left to right illustrates the RGB input, the ground truth depth, the depth estimated by the proposed method, the absolute difference between ground truth and estimated depth, and 3D hand pose estimation results of Pose-REN applied to the ground truth depth and the estimated depth.

Chapter 3
Probabilistic Generative Modeling of Images

There are three popular approaches for achieving generative properties on deep learning models, the generative adversarial networks, diffusion models, and autoencoding variational inference. Generative Adversarial Networks (GANs) [52] have made significant breakthroughs over the recent years, and are capable of generating high-quality images compared to other approaches. However, they suffer from a lack of inference performance, training instability, and mode collapse (i.e. low sampling diversity) [32]. Diffusion models [142] exhibit impressive generative capabilities, especially for image generation, however, they lack interpretability and expressiveness and are computationally demanding. In contrast, VAEs [79] exhibit a stable training procedure, are computationally efficient, are more resilient to mode collapse, have structured latent representation, and are capable of inference, making them prominent in many domains, such as learning disentangled representations and reinforcement learning [32]. While progress has been made towards improving all approaches, the search for better generative models remains an active research field [32, 49, 54, 57, 63, 70, 93, 116, 124, 129, 143, 144, 155, 161, 169].

In this chapter, we propose an improvement (Fig. 3.1) for many VAE architectures. The improvement is based on a modification of the neural network nodes, which are the building blocks of all ANNs. Typically, nodes in neural networks receive activations from their input nodes and perform computations to generate their own activation. Traditionally, each node activation is a single scalar value for a given input to the neural network. When dealing with probabilistic quantities, traditional ANN layers resort to sampling operations in order to approximate them. Such approaches are typically bound to the number of samples drawn (e.g. reparameterization trick in VAEs) limiting their potential expressibility. Therefore, a different approach must be adopted to better exploit the expressiveness of distributions. Toward this end, we propose to generalize traditional approaches by substituting the single scalar-valued activations with probability distributions. Specifically, we assume that each activation has a Probability Density Function (PDF), which we represent

(a) Input image

(b) Original VAE

(c) RV-VAE

Figure 3.1: The proposed RV modifications of VAE architectures enhance the models' capabilities so that a given input image can be reconstructed by the RV-VAE in a way that is perceptually more plausible compared to the original VAE.

by its two first moments, the expected value and variance. Using the algebra of Random Variables [145], we compute the expected value and variance of many commonly used neural network operations given a set of input expected values and variances. In most cases, this can be achieved without any further assumptions, yielding an exact computation. For the case of the ReLU nonlinear operation and Batch Normalization, we make specific assumptions to facilitate the tractability of computations.

Using this mathematical toolbox of Random Variable (RV)-aware operations, we proceed to modify the decoder part of several VAE architectures we tested. This is achieved by replacing all the layers of the decoder parts with our proposed RV-aware layers. By doing so, we alleviate the requirement for the reparameterization trick. Other approaches [32,79] need to implement and use a sampling method to address this. This innovation of the structure of VAEs, leverages the notion of visual inductive priors to reduce data requirements and improve the model's overall data efficiency.

The proposed approach can improve the performance of most existing VAE architectures by replacing the appropriate elements of the network with our proposed ones and retraining it. This modification is easy to implement and can readily enhance image reconstruction quality (MSE) and the fidelity of generative results (Fréchet Inception Distance). Importantly, the proposed approach achieves these results without impacting the training speed in terms of convergence rate. We name this new family of improved modified VAEs as Random Variable Variational Autoencoders (RV-VAEs).

In summary, the contributions of this work are: (a) A theoretically justified approach for utilizing continuous distributions in ANNs and specifically in VAEs for incorporating inductive priors, that (b) significantly improves image reconstruction and (c) achieves similar or improved generative results, while (d) maintaining training convergence rate.

Moreover, (e) we provide the source code[1] for all the RV-aware modules, together with the modifications applied to all the RV-aware VAEs. This work emphasizes the potential of combining deep learning with inductive priors, towards more data-efficient deep learning practices.

3.1 Literature Overview

In this section, we provide a brief overview of the existing literature on enhancing and improving VAEs, as well as works that have applied a stochastic or probabilistic approach to neural networks and autoencoders. Our work fits in both classes with greater emphasis on the former, since it can enhance the performance of most VAE-based architectures.

3.1.1 Enhancing and Improving VAEs

Variational Autoencoder, one of the first successful generative deep learning models, was proposed by Kingma and Welling [79]. Another very successful generative model, Generative Adversarial Network [52] was proposed almost simultaneously. Since VAEs were introduced, a significant amount of work has been devoted to both theoretical analysis [48] and improving the base architecture.

More specifically, Vahdat and Kautz [161] propose a novel architecture for use within the VAE framework, that is shown effective in several tasks using datasets for handwritten digits [83], common objects [80] and faces [72, 89]. Apart from the novel network design, an additional regularization strategy called spectral regularization is also adopted and employed.

Sonderby *et al.* [143] propose a novel VAE architecture, termed Ladder VAE, adapted from the related Ladder Network [123]. Additionally, a data-dependent approximate likelihood is used to correct the generative distribution recursively.

Tomczak and Welling [155] propose a mixture distribution such as a mixture of Gaussians as the prior for the VAE latent space. Furthermore, the problem of inactive stochastic units, typically exhibited in VAE architectures is effectively tackled by adopting a two-layered VAE architecture.

Razavi *et al.* [124] employ auto-regressive priors on the latent space, achieving large-scale, high-coherence and fidelity images. For sample generation, the proposed approach requires sampling an auto-regressive model in the latent space, which is a highly efficient operation since typically the latent space is very small, with its dimensionality not exceeding a few hundred channels.

Kalatzis *et al.* [70] revisit the commonly adopted Gaussian prior over the latent space, replacing it with a Riemannian Brownian motion prior. The authors also propose an infer-

[1]The source code of our method is available at `https://github.com/VassilisCN/RV-VAE`.

ence scheme that is suitable for the employed prior, and show that this results in an overall increase in model capacity.

Ghosh *et al.* [49] propose an alternative regularization scheme for VAEs. The authors introduce an ex-post density estimation step to retrieve a generative mechanism to sample new data. Similarly to our approach, this proposed step can be readily applied to existing VAEs. The authors show that their proposed approach improves the retrieved sample quality.

Xu *et al.* [169] propose a new algorithmic framework for learning autoencoders of data distributions. Specifically, the authors propose to minimize the discrepancy between the model and target distributions, with a relational regularization on the learnable latent prior. This regularization penalizes the Fused Gromov-Wasserstein (FGW) distance between the latent prior and its corresponding posterior.

Huang *et al.* [63] propose an introspective variational autoencoder (IntroVAE) model for the generation of high-resolution photographic images. An inference and a generator model are jointly trained in an introspective way. The authors adopt a scheme similar to GANs, pitting these models against each other. IntroVAE is shown to produce high-resolution photo-realistic images, comparable in quality to that of GANs.

Another hybrid between GANs and VAEs is proposed by Makhzani *et al.* [93]. The proposed model uses GANs to perform variational inference by matching the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior distribution. Doing so ensures that generating from any part of prior space results in meaningful samples. Experiments on several datasets demonstrate the efficacy of the approach.

Han *et al.* [54] propose a joint training method to learn both a VAE and a latent energy-based model. Towards this end, a custom objective function is formulated, using three terms of Kullback-Leibler divergence between three joint distributions on the latent vector and the image. The objective function therefore assumes the form of a divergence triangle that seamlessly integrates variational and adversarial learning.

Pidhorskyi *et al.* [116] explore the issues of whether autoencoder networks have the same generative power of GANs, or learn disentangled representations. The authors design two autoencoders that experimentally exhibit disentanglement properties. Furthermore, one of the proposed architectures trained on a dataset of faces is shown capable of producing face reconstructions and manipulations.

Heljakka *et al.* [57] present a generative autoencoder that provides fast encoding, faithful reconstructions (e.g., retaining the identity of a face), sharp generated/reconstructed samples in high resolutions, and a well-structured latent space that supports semantic manipulation of the inputs.

Finally, Daniel and Tamar [32] propose the Soft-Intro-VAE, a model based on the IntroVAE by Huang *et al.* [63]. Their modified IntroVAE replaces the hinge-loss terms used by the original with a smooth exponential loss on generated samples. This change sig-

nificantly improves the overall performance of the model with state-of-the-art generative results.

3.1.2 Probabilistic ANNs

The tools of probability theory have been heavily used in the field of neural networks, mostly for facilitating the theoretical analysis of their behavior. While vastly used inside ANNs' architectures, most works emphasize on estimating probabilistic quantities rather than employing them within the architecture's core.

Mean-field theory has been applied to the analysis of networks, either on single layers [141] or on multiple layers [31, 96], closely resembling modern deep architectures. Furthermore, Bayesian Neural Network (BNN) embed probabilistic modeling directly into neural architectures [91]. By placing priors on the weights and biases [16] and with the usage of variational inference [148], BNNs introduce a principled uncertainty estimation into deep learning, making them particularly advantageous for tasks where understanding uncertainty is crucial.

On the practical front, mainstream Deep Learning frameworks such as TensorFlow [4] and PyTorch [113] include libraries [14, 34, 138, 158] that facilitate the development and integration of stochastic operations in neural networks. Recently, Kim [78] presented work on the VAE architecture that employs an inference model to enhance the encoding of the data, aiming to reduce the posterior approximation error of inference in VAEs. Kim's work focuses on a more accurate modeling of the computation of latent space values. In contrast, our work focuses on a theoretically justified way of utilizing the encoded latent space.

Within this research area, a particular direction that is closely related to ours is that of Probabilistic Circuits. Poon and Domingos [117] and Shen *et al.* [136] have explored the possibility of developing deep probabilistic models, where the propagation of the PDF throughout the network nodes is constrained to use specific operations, similar to our approach.

Vergari *et al.* [165] compiled a comprehensive list of operations that can be used for the layers of the network toward this end. Additionally, Jaini *et al.* [68] and Cohen *et al.* [30] advocate for the use of tensor decompositions to bring probabilistic models closer to modern deep neural networks. Compared to traditional deep neural networks, probabilistic circuits have some limitations such as their high computational cost and the lack of diversity of their generated samples.

The proposed work can be categorized as a general improvement for VAE architectures, applicable to any VAE architecture that employs sampling in the latent space. The modification is applied only to the decoder part of the architecture, where we substitute the traditional components with our proposed Random Variable-aware ones. To the best of

our knowledge, no similar approaches have been proposed so far in the relevant literature.

3.2 Random Variable Modules

All data that are processed by ANNs such as images, video sequences, audio, and text, are samples of (possibly implicit) underlying distributions. Even though the distribution domains can be infinite, the networks we employ always operate on specific, constant instances belonging to these domains, that is, samples of the distributions. This is the assumption on which we base the design of the operations that comprise an ANN, such as fully connected layers, convolutional layers, activation functions, etc.

A simple, effective and powerful way to represent a potentially infinite range of such instances is to resort to a stochastic representation of the input, a representation that must also be propagated in the same way through the network. If we want the network to process a tensor of non-constant values, such as RVs, we must redefine the network's operations to treat the tensors as such. This can be achieved through the algebra of random variables [145]. Representing network inputs and activations as RVs is very general and powerful, however in practice it quickly leads to intractable computations. In order to come up with a practical solution that can compete with the extremely efficient modern neural networks, some simplifying assumptions need to be made. We adopt those assumptions as we validate them empirically (Sec. 3.2.6).

An arbitrary probability distribution over the real numbers is fully determined by the infinite series of its moments. In this work, we choose to approximate it using only the first two moments, namely the expected value $\mathbb{E}[X]$ and variance $\text{var}[X]$. Throughout the computations performed over the layers, we keep this representation by calculating the new expected value and variance. Another simplifying assumption is the handling of correlations between the involved random variables. Similarly to Batch Normalization [65], for computational efficiency, we choose to ignore all correlations between the inputs to a network layer, and only estimate the variance of each RV. Otherwise, we would have to compute a full covariance matrix, quadratically increasing the need for computations and storage, with respect to the size of each layer.

The rules provided by the algebra for symbolic manipulation are applied in two cases: (a) between two RVs, and (b) between a RV and a constant. Since all commonly used ANN operations are essentially instances of one of these cases, we can adapt and use them to derive all the cases we are interested in. In summary, for a RV X that can be sufficiently described by its mean and variance, and an ANN operation $op(\cdot)$, in order to compute $Y = op(X)$, it is sufficient for our representation to calculate $\mathbb{E}[op(X)]$ and $\text{var}[op(X)]$. It is important to note that, the number of network parameters remains constant, even as the number of operations and network activations increases. In the following sections, we

elaborate on the calculation of expected value and variance for the most common operations of an ANN's modules.

Even though, for the context of this work, we employ our RV modules in specific network modifications (Sec. 3.3) some preliminary results in Appendix B show the beneficial utilization of these modules and in other tasks as well.

3.2.1 Linear operations

General linear operations between the inputs of a neuron are commonly used to implement fully connected layers. Such a linear operation is defined as:

$$\boldsymbol{y} = \mathit{Linear}(\boldsymbol{x}, \boldsymbol{A}, \boldsymbol{b}) = \boldsymbol{x}\boldsymbol{A}^T + \boldsymbol{b}, \tag{3.1}$$

where $\boldsymbol{x}$ is the input vector of RVs, $\boldsymbol{A}$ the matrix of learnable weights, and $\boldsymbol{b}$ the learnable bias vector. The expected value for the output vector of RVs $\boldsymbol{y}$ is given by:

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{A}^T + \boldsymbol{b}] = \mathbb{E}[\boldsymbol{x}]\boldsymbol{A}^T + \boldsymbol{b} = \mathit{Linear}(\mathbb{E}[\boldsymbol{x}], \boldsymbol{A}, \boldsymbol{b}), \tag{3.2}$$

since $\boldsymbol{A}$ and $\boldsymbol{b}$ contain constant values, the expected value $\mathbb{E}[\cdot]$ is only affecting the RV $\boldsymbol{x}$.

We know from the algebra of RVs that summation between a scalar and a RV does not affect the variance of the output RV (i). Also, if a RV is multiplied with a scalar, then the variance of the new RV is multiplied by the square value of the scalar. This can be extended to vector/matrix operations as well. Let $\boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{A}$ where $\boldsymbol{x} = [X_1, \ldots, X_m]$ is a vector of m independent RVs, and $\boldsymbol{A}$ is a $m \times n$ matrix. Consider the RVs:

$$Y_k = \sum_i^m X_i a_{ik} \tag{3.3}$$

for $k = 1, \ldots, n$. Using Bienaymé's identity, and assuming independence of the X_i RVs, the variance equals to the sum of the variances of the summed RVs. Since the multiplication of a RV with a scalar modifies the new variance by multiplying it with the square of that scalar, we can formulate the variance as:

$$\mathrm{var}[Y_k] = \sum_i^m \mathrm{var}[X_i a_{ik}] = \sum_i^m \mathrm{var}[X_i] a_{ik}^2. \tag{3.4}$$

As we can see, each element of the matrix $\boldsymbol{A}$ is squared. This can also be expressed through the Hadamard (element-wise) product (ii). Therefore from (i) and (ii), the variance of Eq. (3.1) can be written as:

$$\mathrm{var}[\boldsymbol{y}] = \mathrm{var}[\boldsymbol{x}\boldsymbol{A}^T + \boldsymbol{b}] = \mathrm{var}[\boldsymbol{x}](\boldsymbol{A} \odot \boldsymbol{A}) = \mathit{Linear}(\mathrm{var}[\boldsymbol{x}], (\boldsymbol{A} \odot \boldsymbol{A}), 0), \tag{3.5}$$

where $\odot$ denotes the Hadamard product.

3.2.2 Convolutional/Transposed convolutional operations

Convolutional operations can be viewed as a special case of general linear operations, where some of the elements of $\boldsymbol{A}$ are forced to be zero. Therefore, the derivation is very similar to the case presented in Sec. 3.2.1. The convolutional operation is defined as:

$$\boldsymbol{y} = Conv(\boldsymbol{x}, \boldsymbol{A}, \boldsymbol{b}) = \boldsymbol{x} * \boldsymbol{A} + \boldsymbol{b}, \tag{3.6}$$

where again $\boldsymbol{x}$ is the input vector of RVs, $\boldsymbol{A}$ the matrix (kernel) of learnable weights, and $\boldsymbol{b}$ the learnable bias vector. The expected value of output $\boldsymbol{y}$ is:

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{x} * \boldsymbol{A} + \boldsymbol{b}] = \mathbb{E}[\boldsymbol{x}] * \boldsymbol{A} + \boldsymbol{b} = Conv(\mathbb{E}[\boldsymbol{x}], \boldsymbol{A}, \boldsymbol{b}) \tag{3.7}$$

and the variance is:

$$\text{var}[\boldsymbol{y}] = \text{var}[\boldsymbol{x} * \boldsymbol{A} + \boldsymbol{b}] = \text{var}[\boldsymbol{x}] * (\boldsymbol{A} \odot \boldsymbol{A}) = Conv(\text{var}[\boldsymbol{x}], (\boldsymbol{A} \odot \boldsymbol{A}), 0). \tag{3.8}$$

A similar procedure is followed to obtain the expected value and variance of the transposed convolution. Since the operation is essentially the same, and only the shape of the kernel is changed between the two operations, the expected value and variance are derived identically to the convolution operation.

3.2.3 ReLU activation function

For the case of the ReLU activation function, the output vector $\boldsymbol{y}$ is defined as:

$$\boldsymbol{y} = max(\boldsymbol{x}, 0) \tag{3.9}$$

In this case, the calculation of the expected value and variance is not straightforward. To calculate the expected value and variance of a RV that is distributed according to the output of the ReLU function, we make the assumption that each input RV follows a normal distribution. This hypothesis is grounded on the empirical observation that after some ANN linear operations, data tend to become approximately normally distributed. This is evidenced in the cases of uniformly and normally distributed data, where the summation of such data is approximately normally distributed as well as for the mixed cases of normal plus normal and normal plus uniform [166]. Specifically, it is distributed according to the Irwin-Hall distribution. This applies to our cases as well, as most of the defined operations perform linear operations over their input. It is also further supported by our own experiments (See Sec. 3.2.6).

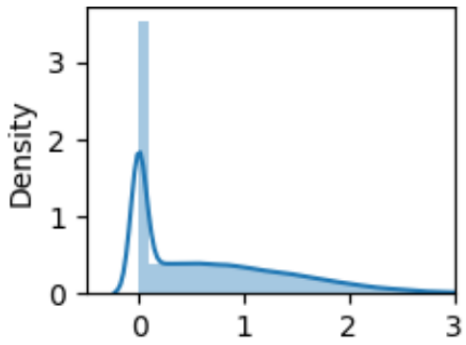


Figure 3.2: Histogram of output values of a ReLU activation function.

Expected value: Using the law of total expectation, we can define the expected value of $max(X, c)$ for a normally distributed RV X and a constant c, as follows:

$$\mathbb{E}[max(X, c)] = \mathbb{E}[X|X > c]P(X > c) + \mathbb{E}[c|X \leq c]P(X \leq c), \tag{3.10}$$

where $P(\cdot)$ denotes the probability function for the provided event. From Eq. (3.10), for the case $c = 0$ of ReLU, $\mathbb{E}[0|X \leq 0]P[X \leq 0] = 0$. Since X is assumed to follow a normal distribution,

$$P(X > c) = 1 - \Phi(a), \tag{3.11}$$

where $\Phi(\cdot)$ is the standard normal Cumulative Distribution Function (CDF) of the normally distributed RV X, and $a = \frac{c-\mu}{\sigma}$ with μ the mean of the normal distribution which is also the expected value, and σ its standard deviation where $\sigma^2 = \text{var}[X]$. The term $\mathbb{E}[X|X > c]$ can be calculated based on the one-sided truncated normal distribution, this can be also observed by looking at the histogram of output values of a regular ReLU activation function Fig. 3.2. Therefore this term can be expressed as:

$$\mathbb{E}[X|X > c] = \mu + \frac{\sigma\varphi(a)}{1 - \Phi(a)}, \tag{3.12}$$

where $\varphi(\cdot)$ is the standard PDF of the normally distributed RV X. By injecting Eqs. (3.11) and (3.12) in Eq. (3.10) we obtain the final form of the expected value as follows:

$$\mathbb{E}[max(X, 0)] = (1 - \Phi(a))\left(\mu + \frac{\sigma\varphi(a)}{1 - \Phi(a)}\right). \tag{3.13}$$

Variance: For calculating the variance, we use the law of total variance in a similar manner:

$$\text{var}[max(X,c)] = \text{var}[X|X>c]P(X>c) + \mathbb{E}[X|X>c]^2(1-P(X>c))P(X>c). \tag{3.14}$$

In this case, the one-sided truncated normal distribution gives us the term:

$$\text{var}[X|X>c] = \sigma^2\left(1 + \frac{a\varphi(a)}{1-\Phi(a)} - \left(\frac{\varphi(a)}{1-\Phi(a)}\right)^2\right). \tag{3.15}$$

Using the Eqs. (3.11), (3.12) and (3.15) in Eq. (3.14) we obtain the final expression of the variance:

$$\text{var}[max(X,0)] = (1-\Phi(a))\left(\sigma^2\left(1 + \frac{a\varphi(a)}{1-\Phi(a)} - \left(\frac{\varphi(a)}{1-\Phi(a)}\right)^2\right) + \left(\mu + \frac{\sigma\varphi(a)}{1-\Phi(a)}\right)^2\Phi(a)\right). \tag{3.16}$$

3.2.4 Batch normalization

As described by Ioffe and Szegedy [65], the batch normalization operation can be broken down into the following two steps: (1) data normalization and (2) data scaling and shifting. The authors state that data normalization is performed for each feature dimension of the data. After normalizing the data they add a linear operation that scales and shifts the data given the learnable parameters γ and β, respectively. Since the second step is essentially a linear operation, it can be handled as in Sec. 3.2.1. On the contrary, the first step needs further analysis, as follows.

We follow the same reasoning stated in Sec. 3.2.3 that all our data become normally distributed after an ANN linear operation. This holds since a batch normalization layer is used only after linear ones. As already stated in Sec. 3.2.3 and proven later in Sec. 3.2.6, all data become normally distributed after some ANN operations. Since each network activation is a RV that follows a distribution, the normalization in the feature dimension can be described as an equally weighted mixture of these distributions in that dimension. Consequently, we want to calculate the mean and the variance of an equally weighted mixture of normal distributions. Toward this end, for a mixture of n component distributions with $X_1, \ldots X_n$ RVs with known expected values and variances, the total mean is defined as:

$$\mathbb{E}[X] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[X_i]. \tag{3.17}$$

The total variance can be calculated as:

$$\begin{aligned} \text{var}[X] &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \\ &= \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[{X_i}^2]\right) - \mathbb{E}[X]^2 \\ &= \left(\frac{1}{n}\sum_{i=1}^{n}(\text{var}[X_i] + \mathbb{E}[X_i]^2)\right) - \mathbb{E}[X]^2 \,, \end{aligned} \tag{3.18}$$

where in the last equality the substitution derives from the fact that $\text{var}[X_i] = \mathbb{E}[{X_i}^2] - \mathbb{E}[X_i]^2$.

3.2.5 Other operations

The operations for which we have derived random variable versions are mainly selected for the use-case of VAEs as we will see later (Sec. 3.3). Therefore, in order to generalize to other architectures, more modules should be studied to form the complete family of ANN operations. A commonly used ANN operation we haven't presented in our derivations in this section, is that of max pooling. Max pooling is commonly used in the encoding part of some VAEs architectures, as well as in many other architectures when down-sampling is required (it can be bypassed by using convolution instead). There is a theoretical difficulty behind this: the PDF of the variable Y where $Y = max(X_1, X_2)$ for some given RVs X_1 and X_2 is generally difficult if not impossible to derive analytically. For example, if X_1 and $X_2 \sim \mathcal{N}(0,1)$ then the CDF of Y is $\Phi(y)^2$, which doesn't have an analytic integral. Things get only more complicated if the two distributions are not identical (they belong to different families etc). In practice, it is feasible to reasonably approximate the expected value and variance of Y for given input distribution expected values and variances. However, there is no exact analytical formula for the general case.

Other operations that do not affect the data values but only transform the expected value and variance linearly according to the respective operation. Such operation is Up-/Down-sampling. This family of operations does not require some form of special treatment. Therefore, for such an operation $op(\cdot)$, the output RV Y for an input RV X has an expected value of:

$$\mathbb{E}[Y] = op(\mathbb{E}[X]), \tag{3.19}$$

and a variance of:

$$\text{var}[Y] = op(\text{var}[X]). \tag{3.20}$$

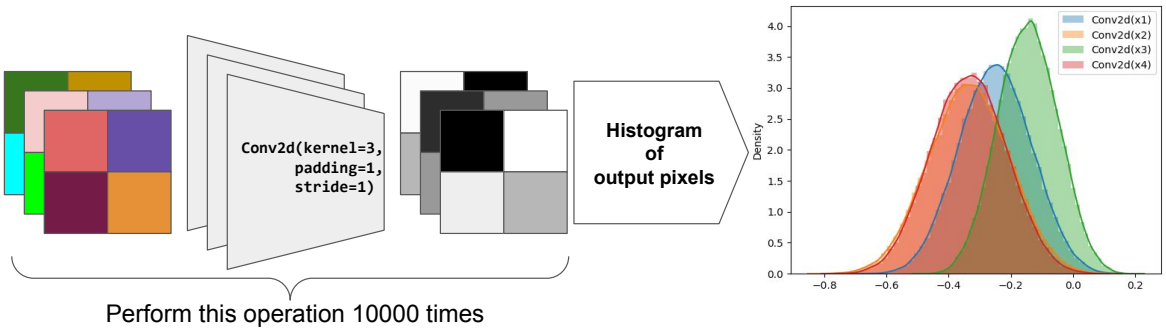


Figure 3.3: The histogram of 4 output pixels after a convolution operation. The histogram was obtained by repeating the same process for 10, 000 times. The input pixels were sampled each time from $\mathcal{U}(0, 1)$.

3.2.6 Normally Distributed Data Assumption

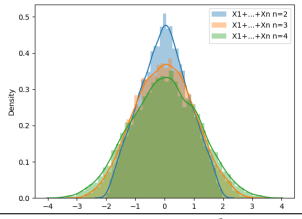
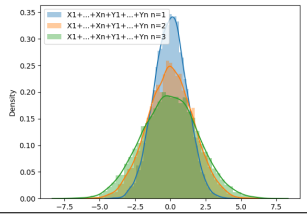
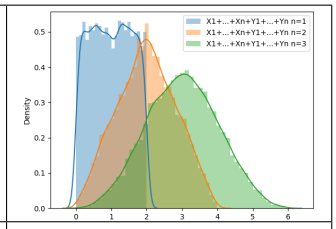
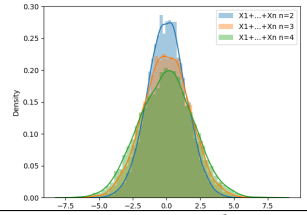
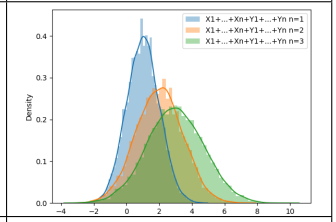
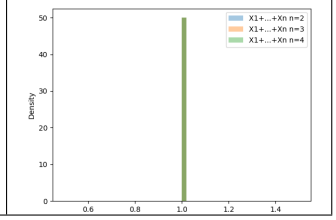
As stated before, all ANN operations can be adapted for the case of RV input using the rules of random variable algebra. In order to comprehend how the rules apply to our case, we must clarify that for every ANN operation, the input to the operation is a RV, while the parameters (weights and biases) are scalars. That is the reason why in this section (Sec. 3.2), all modifications consider the case of a RV operand and a scalar. Moreover, due to the way these operations are defined, each RV is only summed with other RVs (never multiplied). This observation lays a solid basis for proving the assumption we make regarding normal data distribution.

The assumption of normally distributed data was used for calculating the expected value and variance of ReLU activation function (Sec. 3.2.3) and batch normalization layer (Sec. 3.2.4). This assumption is supported by the following empirical evidence.

We conducted a series of experiments in order to evaluate the distribution of data after some ANN operations. This data is provided as input to a non-linear activation function. All the above operations involve linear combinations of different operand types. More specifically, the operands are of types: Uniform, Normal, and Scalar. Therefore, we only need to show that summations of different operand types result in Normal-like distributions.

Table 3.1 shows the histograms of sums for different combinations of X and Y operands for different quantities of operands. The histograms show the density of 10, 000 sampled values, where each sample is the sum of n operands. For every type of operand, if an RV X is of type Uniform then $X \sim \mathcal{U}(-1, 1)$, if X is of type Normal then $X \sim \mathcal{N}(0, 1)$, if X is of type Scalar then $X \sim \mathcal{N}(1, 0)$, therefore for each sample each operand is drawn/sampled from its respective distribution. As we can observe, in most cases the resulting histograms

Table 3.1: Histograms of summations for combination of different operands, where n is the number of operands.

Operand	Y = Uniform	Y = Normal	Y = Scalar
X = Uniform			
X = Normal	symmetric plot		
X = Scalar	symmetric plot	symmetric plot	

take a Gaussian form even for a few operands. If all operands are of scalar type, we notice that, as expected, the output distribution is a Dirac delta function. We can describe this distribution in a Gaussian-like form as $\mathcal{N}(m, 0)$ where m is the output scalar. In the case of uniform plus scalar operands, we can observe that the output distribution is similar to an Irwin-Hall distribution, which is the case of uniform plus uniform. Therefore, for a large number of operands (specifically $n > 2$) the output yields a Gaussian-like distribution (third "green" histograms of Tab. 3.1).

Furthermore, we performed Kolmogorov-Smirnov tests on all cases presented in Tab. 3.1 (for more than 2 operands). Specifically, we tested the null hypothesis that the samples after each operation are distributed according to a Normal distribution. During these tests, the p-value ranged from 0.158 to 0.926 (for lower to higher number of operands). In all cases, the p-value was larger than 0.05, thus we could not reject the null hypothesis.

More empirical results are reported in Fig. 3.3. For an input noise image where (the value of) each pixel is sampled from $\mathcal{U}(0, 1)$, we show the histogram of the output pixel values (4 pixels) after a convolution operation, repeated for $10,000$ times. Even though the input is uniformly distributed, the output pixel values follow a Gaussian-like distribution

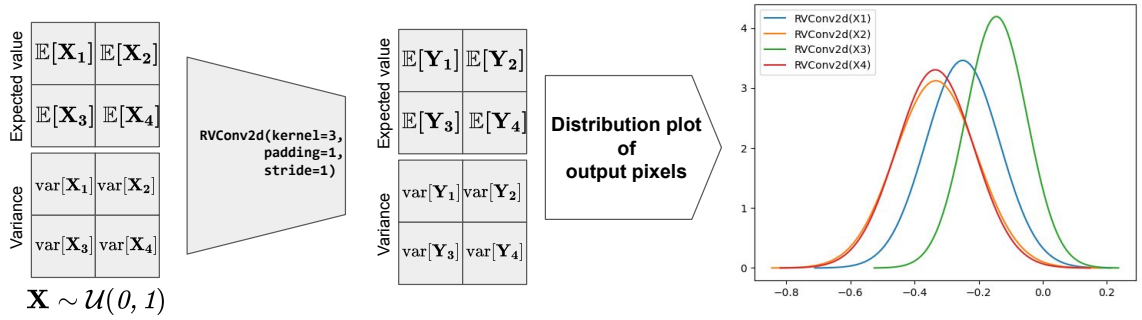


Figure 3.4: The plot of 4 Normal distributions created from the 4 output pixels after a single RV-convolution operation. The means and variances of those 4 Normal distributions are the respective means and variances of the output pixels'. The input values were RVs with expected value and variance $\mathbb{E}[\mathcal{U}(0,1)]$ and $\text{var}[\mathcal{U}(0,1)]$ respectively.

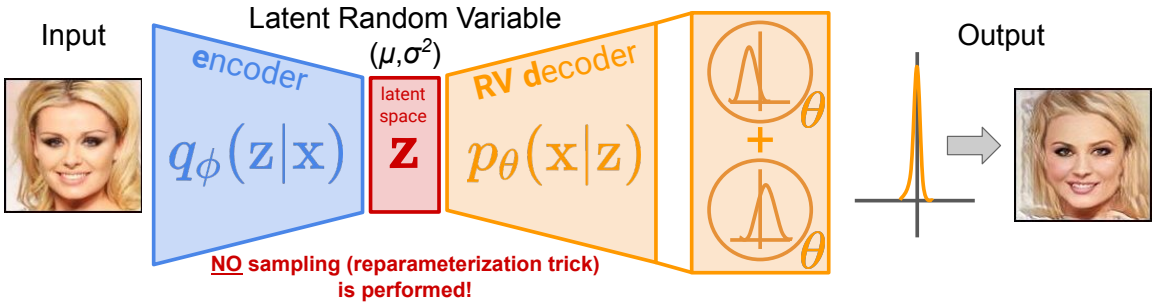


Figure 3.5: The proposed VAE formulation avoids the need for stochastic sampling from the latent space variables $\boldsymbol{z}$, by directly forwarding the encoded distributions $q_{\boldsymbol{\varphi}}$ to the decoder. This is achieved by treating the latent space as an instance from a family of distributions and employing random variable operations inside the decoder. The final output is also a distribution and, by minimizing its variance, we effectively constrain it to become a constant (image). Following standard VAE notation (as in Kingma and Welling [79]), $q_\varphi(z|x)$ and $p_\theta(x|z)$ denote the encoder and decoder part of the network respectively, and vectors φ, θ and x denote respectively the parameters of the encoder, decoder, and the input.

after a single convolution operation.

Figure 3.4 depicts the same operation as above but performed in a single RV convolution as defined in Sec. 3.2.2. For an equivalent input RV matrix, we show the plot of 4 Normal distributions created after a single RV convolution operation. We can observe that the distributions created are nearly identical to the histograms of Fig. 3.3.

3.3 Random Variable Variatonal Autoencoders

The building blocks described, are capable of handling RVs, and can be used to replace stochastic procedures in ANNs. A suitable group of architectures is VAEs, which feature a stochastic process using scalar value samples. The stochastic usage of latent encoded data distributions at the bottleneck layer that forwards samples of these distributions to the decoder, makes them suitable to be used with RV modules. Figure 3.5 depicts a concise visualization of our proposed modifications in the VAE formulation. By design, the encoded latent space in most VAE architectures is normally distributed, and consequently is able to use RVs of appropriate distributions to represent it. Therefore, we can adjust VAE architectures to incorporate RV modules.

3.3.1 Relation to the original VAE formulation

Our goal in this work is to avoid imposing a stochastic estimation in the lower bound. To better understand this proposed contribution in the VAE approach, we first present here the variational lower bound or Evidence Lower Bound (ELBO) as defined in the original VAE formulation by Kingma and Welling [79]. The evidence lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}^{(i)})$ is defined for a single input data point $\boldsymbol{x}^{(i)}$ as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x}^{(i)})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\mathbf{z})]. \tag{3.21}$$

The formulation follows the standard VAE notation of [79]. $q_\varphi(z|x)$ and $p_\theta(x|z)$ denote the encoder and decoder part of the network respectively, and vectors φ, θ and x denote the parameters of the encoder, decoder, and the input.

The formulation's goal is to differentiate and optimize this lower bound w.r.t. $\boldsymbol{\varphi}$, the variational parameters, and $\boldsymbol{\theta}$, the generative parameters. In order to estimate the second term, the authors suggest forming Monte Carlo estimates of expectation, where the lower bound is estimated as:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}^{(i)}) \simeq \tilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\mathbf{z}^{(i,l)})), \tag{3.22}$$

where L is the number of samples drawn. The authors state that $L = 1$ is sufficient since the training procedure is performed in batches of satisfactory size.

In this work, instead of adopting a stochastic approach in the estimation of the lower bound that also depends on the batch size, we employ the whole distribution $q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})$, according to which the RV $\mathbf{z}$ is distributed, as seen and in Fig. 3.5. Therefore, by implementing differentiable RV operations during the generative process, the method we propose is closely related to the original lower bound formulation of Eq. (3.21), using this term di-

rectly during training, instead of the commonly used approximation Eq. (4.2).

3.3.2 Architecture modifications

The modifications needed to make any VAE-based architecture appropriate for handling RVs are mainly on the decoder of the architecture. Since the encoder outputs the latent distribution (in the form of means and variances), in an unmodified VAE architecture we would forward it through the decoder via sampling. In our case, we do not need to perform any sampling operation, therefore we operate on the expected values and variances resulting from the encoder, and defer the conversion of distributions to values until the end of the pipeline. Any VAE-based architecture that decodes the encoded distribution can take advantage of the proposed modification. Every layer/module of the decoder must be replaced with the appropriate RV module described in Sec. 3.2. Specifically, the decoder of a VAE consists in most cases of linear modules (such as fully connected layers and convolutional/transposed convolutional layers Sec. 3.2.1), ReLU activation functions (Sec. 3.2.3) and batch normalization layers (Sec. 3.2.4).

3.3.3 Loss adjustment

Since every module outputs RVs (represented by their means and variances), by doing the above modifications in the decoder, the final output of the network will also be a RV. This is not useful in our test cases, but can be accommodated with the following adjustment in the loss function. Provided that the goal of a VAE is to output a result of a constant form (e.g. an image), we can make the modified VAE architecture achieve this by enforcing a constraint on the output RVs of the final layer. A constant can be expressed as a RV in the form of a Dirac delta function, with expected value the constant itself and variance equal to zero. The network's output can be viewed as a constant value if we retain only the mean value and enforce the variance to be close to zero. This can be achieved by minimizing the following term:

$$L_c(\boldsymbol{y}) = \frac{\lambda}{n} \sum_{i=1}^{n} \text{var}[\boldsymbol{y}_i]^2, \tag{3.23}$$

where $\boldsymbol{y}$ is the final output vector of n RVs of a VAE and λ an experimentally determined constant scale factor that acts as a balancing weight between the terms. The final loss is therefore defined as the sum of the original variational lower bound and the new term L_c:

$$L(\boldsymbol{x}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}) + L_c(\boldsymbol{y}). \tag{3.24}$$

3.4 Experimental Evaluation

The conducted experiments aim to verify and assess the following goals of our approach: (1) the ability and ease to adopt RV-awareness by different VAE-based architectures, (2) the improvement in image reconstruction , (3) the improvement of generative capabilities, while (4) maintaining satisfactory or even improved training convergence time.

To tackle these goals we used four different VAE-based architectures, i.e., the original VAE [79], β-TCVAE [24], Soft-Intro-VAE [32] and DC-VAE [111]. The first two are very popular VAE approaches, whereas the latter two are recent state-of-the-art approaches. These methods were trained and tested on several datasets, each using the respective implementation by the authors wherever possible. The VAE and β-TCVAE architecture implementations were used from [147]. Our evaluation employed the following datasets: MNIST [83], CIFAR-10 [80], CelebA [89] and CelebA-HQ [72] resized to 128×128.

3.4.1 Constructing RV-aware VAE architectures

Using the modifications of Sec. 3.3.2 we created an RV-based version of all the above architectures. As mentioned, the modifications were on the decoder and the loss function. In practice, in all architectures we omitted the reparameterization trick, and sent the encoded distributions directly into the respective decoders.

The loss adjustment described in Sec. 3.3.3 was employed by all architectures and was added to their originally defined losses. Specifically, for every output RV tensor $\mathcal{Y}$, the $\mathbb{E}[\mathcal{Y}]$ was responsible for minimizing the reconstruction error, while var$[\mathcal{Y}]$ was used for the added loss L_c described in Eq. (3.23), with $\lambda = 50$ for all the experiments. All architectures were trained using the proposed hyper-parameters in their respective manuscripts and provided code and trained for 150 epochs.

3.4.2 Implementation details

We implemented the modules described in Sec. 3.2 using PyTorch [113]. We used implementations (also in PyTorch), of all the architectures we experimented with, and replaced the appropriate network layers with their RV-enabled respective ones. In practice, we found that our modified VAE architecture implementation is about 3 times slower. However, it only required approximately 30% more Floating Point Operations per Second (FLOPS) than the unmodified version. Therefore, we believe that suitable optimizations, such as implementing demanding modules like ReLU and Batch Normalization in C++, can significantly increase the overall speed of our implementation. More specifically, we expect the speed to improve to around 70% of the unmodified version, as judged by the number of additional FLOPS required by our approach. Overall, our experiments demonstrate that the benefits of our approach in terms of improved reconstruction error and

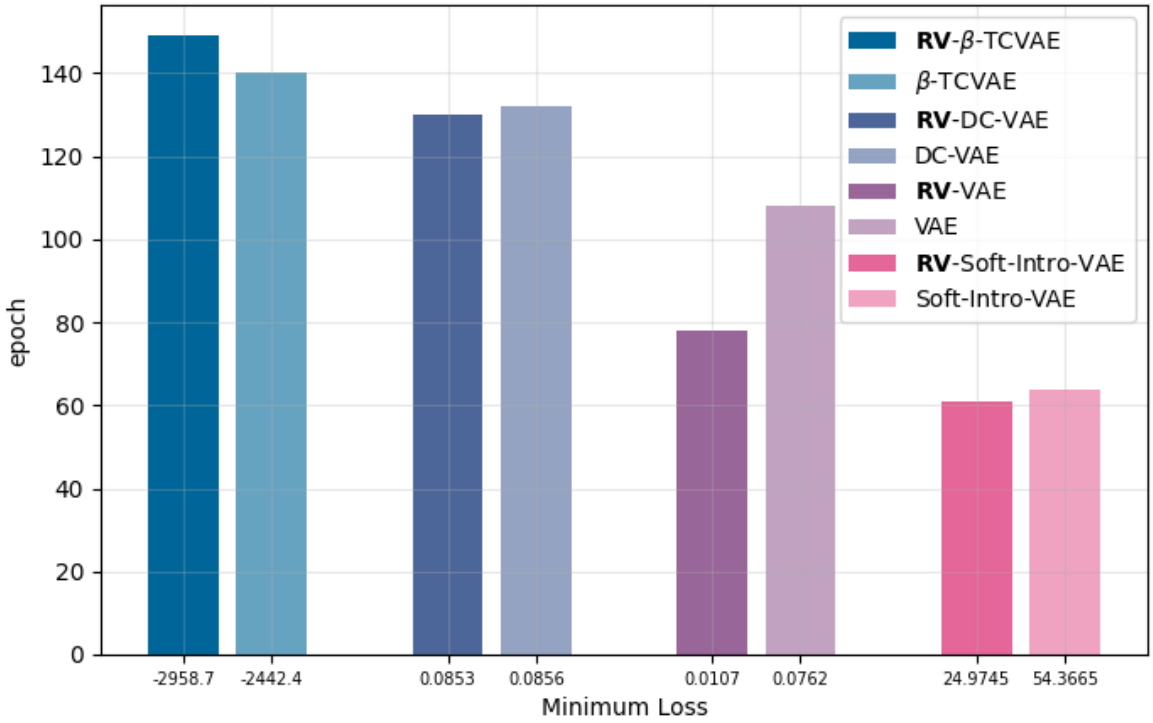


Figure 3.6: The epoch of training (as bar height) that each architecture reached its minimum validation loss value on the CIFAR-10 [80] dataset.

generative capabilities justify the additional computational cost.

3.4.3 Training speed convergence comparison

With the elimination of sampling in the latent space during training, an RV-based VAE network does not rely on thoroughly sampling the training data space, since the whole data distribution is being forwarded to the decoder. This can lead to faster convergence. This is documented in Fig. 3.6 which illustrates the time (in epochs) needed for each VAE architecture to reach its minimum validation loss during training. In all cases, the RV-aware architectures reach their minimum loss in a similar or earlier epoch than their original counterparts. Moreover, the minimum loss of RV-based networks is always, by far, lower than the loss of the original architectures, despite the fact that the loss of RV-aware VAEs includes the additional, non-negative term, L_c.

Table 3.2: Image reconstruction results for all datasets.

	MSE ↓ on MNIST		**MSE** ↓ on CIFAR-10		**MSE** ↓ on CelebA		**MSE** ↓ on CelebA-HQ	
Method	Original	RV-VAE (ours)	Original	RV-VAE (ours)	Original	RV-VAE (ours)	Original	RV-VAE (ours)
VAE [79]	0.0081	**0.0005**	0.0763	**0.0107**	0.0478	**0.0192**	-	-
β-TCVAE [24]	0.0021	**0.0004**	0.0412	**0.0157**	0.0412	**0.0130**	-	-
DC-VAE [111]	-	-	0.1245	**0.1139**	-	-	-	-
Soft-Intro-VAE [32]	0.0194	**0.0129**	0.0211	**0.0155**	-	-	0.0247	**0.0151**

Figure 3.7: Reconstructions of CelebA-HQ [72] images (1st row) by Soft-Intro-VAE [32] (2nd row) and RV-Soft-Intro-VAE (3rd row).

3.4.4 Image reconstruction

To demonstrate the reconstruction capabilities of the RV-aware VAEs, we conducted several experiments comparing the original architectures with our RV-aware modified ones on the employed datasets. Table 3.2 shows the MSE for all test sets of datasets between the original images and their reconstructed ones. In all cases, our proposed RV modifications enhance the reconstruction performance of all the reported architectures, even by a large margin in some cases. To further illustrate those results in a qualitative context, we also provide some reconstructed test samples in Figs. 3.7 to 3.15.

3.4.5 Image generation

The proposed RV modifications are also beneficial due to their generative properties. To illustrate this, we report in Tab. 3.3 the Fréchet Inception Distance (FID) based on $50,000$ generated samples. For generating new samples, we follow the same procedure as in the original VAEs by sampling the mean from a Gaussian distribution and fixing the variance to $\text{var}[X] = 1$. For all cases, we observe lower FID in the modified RV-aware networks. We can also see in Figs. 3.16 to 3.20 some qualitative results of RV-aware generated samples compared to the samples generated by the unmodified networks. Moreover, to show the continuity of the latent space, in Fig. 3.21, we present generated images that are created

(a) VAE [79] (b) β-TCVAE [24]

Figure 3.8: 1st rows: CelebA [89] images; 2nd, 3rd rows: reconstructions by original VAEs and their RV-aware versions.

Table 3.3: Comparison of FID scores for CIFAR-10 [80] and CelebA-HQ [72] datasets. *FIDs calculated by the implementations provided by the authors.

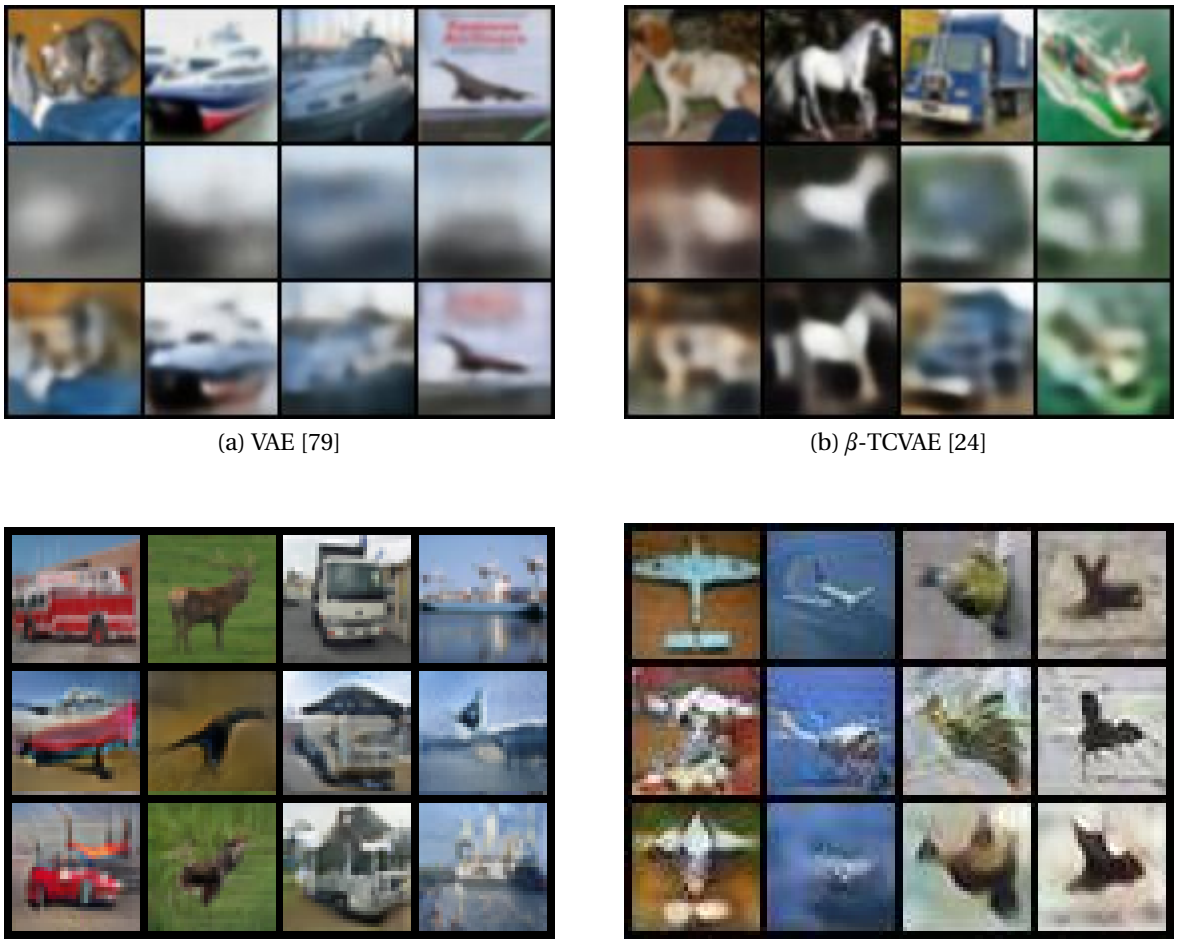
	FID ↓ CIFAR-10 [80]		**FID ↓** CelebA-HQ [72]	
Method	Orig.	RV-VAE (ours)	Orig.	RV-VAE (ours)
DC-VAE [111]*	26.78	**23.44**	-	-
Soft-Intro-VAE [32]*	5.31	**5.26**	2.85	**2.82**

by interpolating between two latent space samples.

3.4.6 Transferability (from RV-VAE to regular VAE)

Despite the proposed changes, the trainable parameters of the resulting, RV-aware networks remain the same. Therefore, after training, it is conceivable to consider the same network weights transferred to a non-RV counterpart. This should be expected to operate without any changes since the Expected Values of the involved quantities behave linearly: For scalar parameters a and b, a relation $Y = aX + b$ between two RVs X and Y implies that the Expected Value of their samples is similarly related, $\mathbb{E}[y \sim Y] = a\mathbb{E}[x \sim X] + b$. Therefore, we can transfer the learned parameters of an RV-VAE network to a regular VAE and keep its functionality.

To provide evidence of this, in Fig. 3.22 we present some examples of reconstructed images. Specifically, we can observe that the third row which presents the reconstruc-



(a) VAE [79]

(b) β-TCVAE [24]

(c) DC-VAE [111]

(d) Soft-Intro-VAE [32]

Figure 3.9: 1st rows: CIFAR-10 [80] images; 2nd, 3rd rows: reconstructions by original VAEs and their RV-aware versions.

tions from a non-RV network that had its parameters transferred from a RV-aware one, has similar results to the last row which has the reconstructions of the original RV-aware network. Moreover, as stated previously these results are significantly better than the ordinary trained non-RV network, second row. This is also justified by the MSE between the RV-aware reconstruction and the reconstruction of transferred RV parameters to a non-RV network to be 1.85×10^{-6}.

Apart from the theoretical feasibility and the experimental validation, it is also useful to address the reason to follow such an approach. Training an RV-aware network is demonstrably beneficial as already presented. Furthermore, evaluating using a *non*-RV network is computationally faster since the computation of the variances in each layer is no longer

Figure 3.10: Reconstructions of CelebA [89] real images (1st row) using original VAE [79] architecture (2nd row) and the proposed RV-VAE version (3rd row).

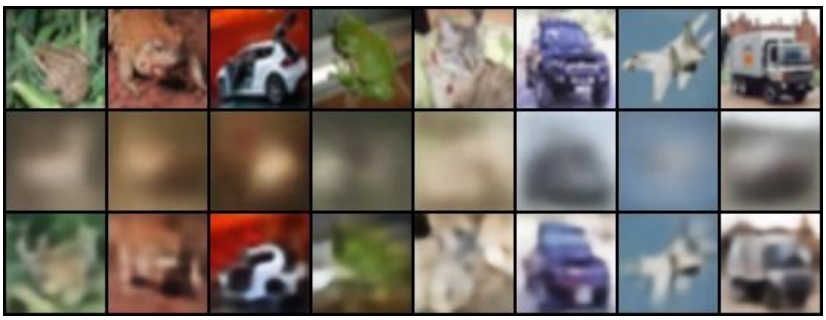


Figure 3.11: Reconstructions of CIFAR-10 [80] real images (1st) using original VAE [79] architecture (2nd row) and the proposed RV-VAE version (3rd row).

necessary. Essentially, the proposed approach acts as a regularization technique that enables better/more accurate results.

3.5 Summary

In this chapter, we presented an approach to enhance a family of probabilistic generative models characterized by well-structured latent spaces: Variational Autoencoder. The introduced improvements focus on probabilistic enhancements aimed at enabling conditional traversal of the latent space, providing greater control over the image generation process. We achieve this enhancement by incorporating continuous distributions into VAE architectures using the algebra of random variables to treat decoder node activations as distributions. This modification can be readily applied to most VAE architectures by simply replacing decoder layers with RV-aware ones, followed by retraining.

Our novel probabilistic framework diverges from traditional sampling-based approaches, yielding improvements in both reconstruction quality and generative fidelity

Figure 3.12: Reconstructions of CelebA [89] real images (1st row) using original β-TCVAE [24] architecture (2nd row) and the corresponding proposed RV-aware version (RV-β-TCVAE version, 3rd row).

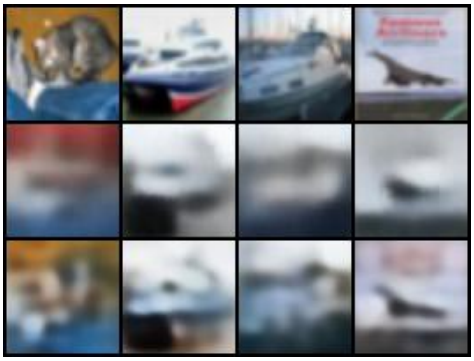


Figure 3.13: Reconstructions of CIFAR-10 [80] real images (1st row) using original β-TCVAE [24] architecture (2nd row) and the corresponding proposed RV-aware version (RV-β-TCVAE version, 3rd row).

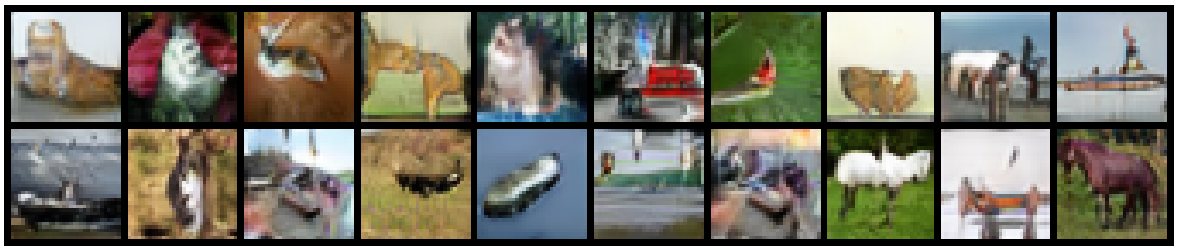
without affecting the convergence rate. The resulting RV-VAEs represent a new class of probabilistic generative models that excel in structured latent space representation and exhibit enhanced generative performance.

The RV-VAE group of models is particularly suited for advanced hand image synthesis in the RGB domain. The well-defined latent space ensures high correspondence between latent vectors and generated outputs. Furthermore, the improvements provided by RV-aware models enable generative capabilities that approach those of larger, more complex models, such as GANs and diffusion models but with fewer parameters. This allows for precise manipulation of the latent space for *conditional* hand image generation which we will elaborate on in the next chapter.

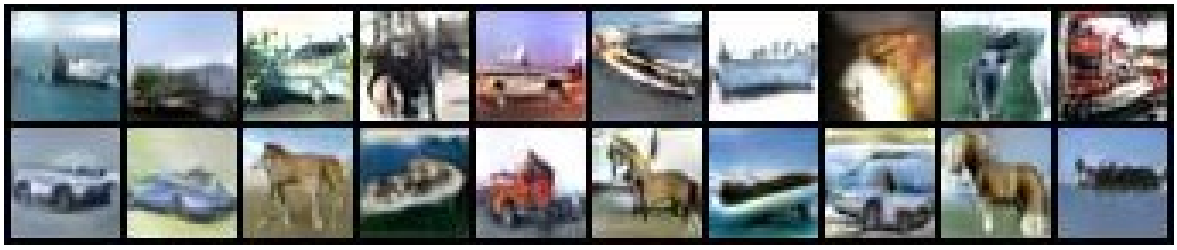
Figure 3.14: Reconstructions of CIFAR-10 [80] real images (1st row) using original Soft-Intro-VAE [32] architecture (2nd row) and the corresponding proposed RV-aware version (RV-Soft-Intro-VAE version, 3rd row).

Figure 3.15: Reconstructions of CelebA-HQ [72] real images (1st row) using original Soft-Intro-VAE [32] architecture (2nd row) and the corresponding proposed RV-aware version (RV-Soft-Intro-VAE version, 3rd row).

Figure 3.16: Generated samples on CelebA-HQ [72] using original Soft-Intro-VAE [32] (1st row) and our RV-Soft-Intro-VAE (2nd row).

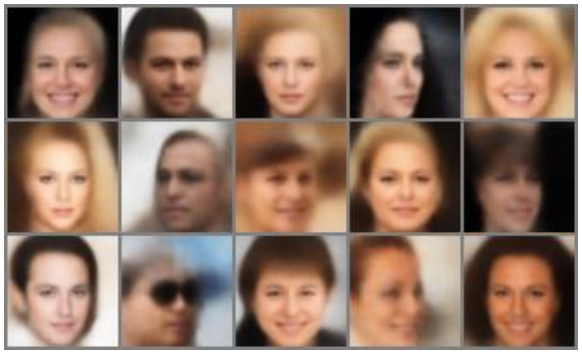


(a) DC-VAE [111]



(b) Soft-Intro-VAE [32]

Figure 3.17: Generated samples on CIFAR-10 [80] using original VAEs (1st rows) and their RV-aware versions (2nd rows).

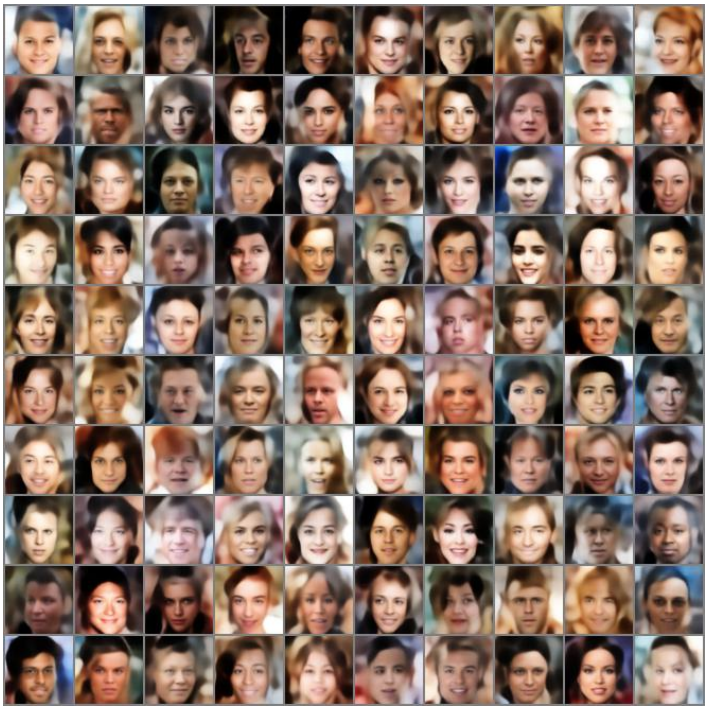


(a) VAE [79]

(b) RV-VAE (ours)

Figure 3.18: Generated samples on CelebA [89] using (a) original VAE [79] and (b) RV-VAE (ours).

(a) VAE [79]



(b) RV-VAE

Figure 3.19: Image generations on CelebA [89] with (a) VAE [79] and (b) the corresponding proposed RV-aware version RV-VAE.

(a) Soft-Intro-VAE [32]

(b) RV-Soft-Intro-VAE

Figure 3.20: Image generations on CelebA-HQ [72] with (a) Soft-Intro-VAE [32] and (b) the corresponding proposed RV-aware version RV-Soft-Intro-VAE.

Figure 3.21: Interpolated generations between two CelebA-HQ [72] samples from RV-Soft-Intro-VAE.

Figure 3.22: 1st row: input images; 2nd row: the reconstructions of input images from an ordinary trained non-RV network; 3rd row: reconstructions from a non-RV network with RV-trained parameters; 4th row: the reconstructions from the RV-aware network.

Chapter 4
Conditional Probabilistic Generation of Images

The accurate estimation of hand pose, shape, and appearance from visual data, as well as the related problem of generating realistic hand images are challenging tasks. As already mentioned in previous chapters, they are widely used in fields such as VR, HCI, and robotics. And a critical aspect of advancing these applications lies in the availability of large-scale, high-quality datasets that can be used to train robust machine learning models. To address the issues of data quality, diversity, and annotation consistency, synthetic data generation has emerged as a viable alternative, providing a scalable solution to augment real-world datasets and enhance model performance.

In recent years, generative models have garnered significant attention for their ability to create realistic data samples that closely resemble real-world data. Techniques such as GANs and VAEs have demonstrated remarkable success across various domains, particularly in image generation. Despite these advances, generating realistic hand images remains a challenging task due to the complexity of hand anatomy and the need for precise control over hand poses. While GAN-based methods like GANerated Hands [98] have made strides in synthesizing diverse hand poses, diffusion models have struggled with generating anatomically accurate hand images, often resulting in distorted outputs.

To overcome these challenges, in this chapter we introduce a novel approach using a Supervised Random Variable Variational Autoencoder (SRV-VAE) for the generation of realistic hand images conditioned on specific hand poses. Our method builds upon the strengths of RV-VAE [102] (Chapter 3), which has been shown to effectively encode and take advantage of complex data representations by directly utilizing the entire distribution of the latent space, leading to improved performance in image generation tasks. By incorporating supervision into a part of the latent space for the pose and leaving the other part unsupervised for the appearance, our approach allows for fine-grained control over hand image generation. This is achieved by combining an appearance vector with a pose configuration, ensuring that the synthesized images are both realistic and pose-accurate

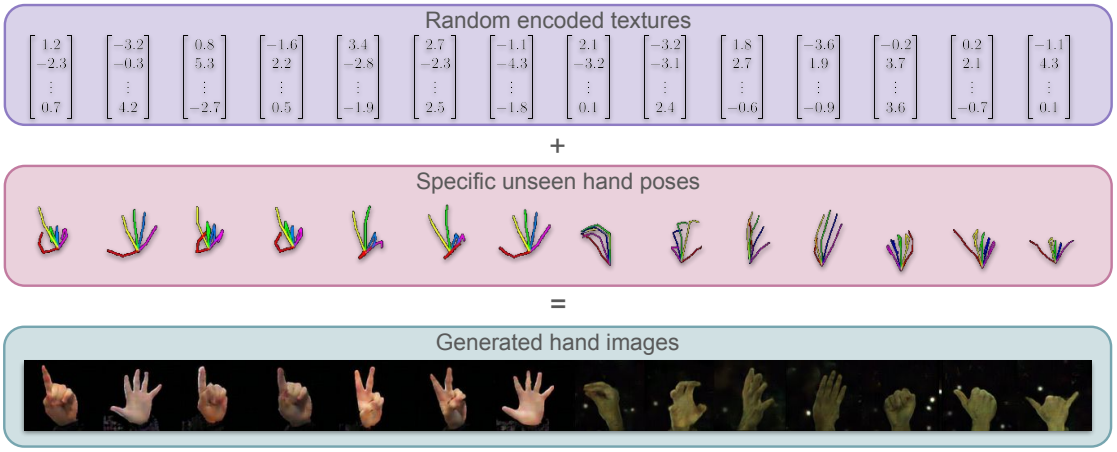


Figure 4.1: By utilizing the normally distributed unsupervised latent texture space and the supervised hand pose space, our method is capable of generating realistic hand images even on unseen 2D/3D hand poses.

even on unseen hand poses, as depicted in Fig. 4.1.

We present a comprehensive evaluation of our SRV-VAE framework, demonstrating its effectiveness in generating high-quality hand images across various poses. We compare our approach against existing methods, highlighting its advantages in terms of both qualitative and quantitative metrics. Additionally, we explore the utility of our generated images in enhancing the performance of hand pose estimation models, providing evidence of the broader applicability of our approach in augmenting hand image datasets.

Overall, our contributions are threefold: 1) We introduce SRV-VAE, a novel generative model that enables controlled and realistic hand image generation by disentangling hand pose and appearance in the latent space. 2) We demonstrate the effectiveness of SRV-VAE in generating high-quality hand images, even for previously unseen poses, by conducting extensive experiments on benchmark datasets. 3) We show that the generated images can be used to enhance the performance of hand pose estimation models, illustrating the practical value of our approach in real-world applications. By addressing the challenges of hand image generation with SRV-VAE, this work contributes to the ongoing efforts to improve hand pose estimation models and provides a robust framework for generating synthetic hand images with precise pose control.

4.1 Literature Overview

Research on visual hand pose, shape, and appearance estimation, as well as the related problem of image synthesis given this information, has progressed in strides in recent

years. Significant contributions that aided advance the field include the introduction of depth sensors [179], the deep learning revolution [82], and the availability of large, relevant, high-quality datasets [28, 98]. The state of the art today relies on increasingly accurate and high-quality data to train better systems on [178]. This demand has led to bigger and better real-world datasets [97], but also to the introduction of synthetic data generation approaches to bootstrap or even complement training [85, 87].

Generative models have drawn significant attention in recent years with their improving capabilities in creating new data samples that resemble real-world data. With the advent of deep learning, architectures such as GANs and VAEs, have achieved remarkable success across various domains, including image generation, natural language processing, and music composition. For instance, GANs have been utilized to generate realistic images from textual descriptions [52, 121]. Similarly, VAEs have shown proficiency in generating high-quality data by learning latent representations [79]. Specifically for image generation, one of the latest and most successful approaches involves diffusion models, which progressively denoise an initially random image to produce high-quality outputs [59, 144]. The evolution of these models has led to remarkable instances, such as the DALL-E [122] series, Stable Diffusion [129], and MidJourney [3], which generate high-quality, customizable images from textual descriptions, pushing the boundaries of generative models [128].

Despite this remarkable progress, diffusion models for images are notoriously bad at generating realistic hands, often producing distorted or anatomically incorrect results. Moreover, accurate hand pose annotation is crucial for generating synthetic data to be used for training machine learning models. Therefore, specialized techniques are still required to overcome these challenges and achieve controllable, precise, high-quality hand image generation.

4.1.1 Hand generative models

The need for precise hand pose representation in synthesized, high-quality images has led to the development of dedicated generative models specifically tailored for hand images. "GANerated Hands" by Mueller *et al.* [98] is one of the early successful works on this topic. By leveraging GANs, the approach creates realistic hand images that can be used for training pose estimation systems. The method addresses the scarcity and limitations of existing hand datasets by generating a diverse array of hand poses and appearances that enhance the robustness of hand pose estimation models. While this method can produce realistic hand images, it requires a whole synthetic hand image instead of just one pose configuration. Moreover, this method makes synthetic hand images more realistic and does not add more variance to the texture/appearance of the generated hand images.

Further refining the capabilities of GANs in the domain of hand image synthesis, H-GAN, introduced by Oprea *et al.* [108], adopts a cyclic consistency approach [180] that

improves the generation process and manages to focus separately on different aspects of the hand, such as preserving the pose of a given image, and altering the texture. While this approach allows for the creation of realistic looking images, it relies on 3D rendering to do so, whereas our proposed approach allows for the generation of new samples directly from real-world data by having control over the pose and the variance of the appearance.

Achieving realism in hand image synthesis requires careful consideration of lighting and illumination, a challenge that Chen *et al.*tackle in their work URHand [25]. Their method enhances the realism of synthesized hand images by generating specific poses under varying lighting conditions. However, it maintains a consistent base texture for the hand, which limits the diversity in the appearance of the generated hands.

In the context of diffusion models, which have typically struggled with the complex structures of hands, the work of Yang *et al.* [172] introduces an innovative approach to improve the quality of synthesized hand images. By incorporating hand pose annotations and focusing on the accurate portrayal of hand anatomy, this method aims to overcome the common distortions and inaccuracies encountered in standard diffusion model outputs.

Another diffusion-based approach is HanDiffuser [100] which generates realistic images of hands in scenes described by a text prompt. This approach uses the pose of a hand which is extracted from an estimated full-body pose, as an intermediate supervision to generate the final image. The main input to this method is text and does not offer precise control over the poses.

Methods that are used with differentiable rendering approaches are HTML by Fu *et al.* [120] and similarly Handy model [118]. Both methods disentangle the texture from the mesh of the hand, however, are not suitable for direct image synthesis since both require complex pipelines to estimate lighting conditions and the background.

4.1.2 Supervision in generative models

The incorporation of supervision techniques into generative models has proven to be a powerful strategy for enhancing model performance and reliability. One of the notable advancements in this area, for the related problem of controlling the body pose of a depicted human, is ControlNet by Zhang *et al.* [178], which integrates supervised learning to guide diffusion models in the image generation process. The resulting method is suitable for tasks requiring accurate control of human poses. While this method can produce realistic images that contain hands given a 2D pose, it requires the whole body pose and struggles to generate images that are focused on hands explicitly.

Supervised approaches have also been extended to autoencoders and VAEs. For instance, the work by Le *et al.* [81] introduces supervision into autoencoders to enhance learning efficiency and output consistency. This can be particularly beneficial in applica-

tions such as medical image analysis.

In VAEs, Berkhahn *et al.* [13] demonstrate how supervised learning can be utilized to enforce specific properties in the generated images, further enhancing the utility of VAEs in complex image generation tasks like hand pose estimation.

Additionally, the integration of supervision techniques has proven effective in anomaly detection, as shown by Kawachi *et al.* [73]. Their approach uses supervised learning to refine the model's ability to identify and differentiate normal from anomalous patterns, which is crucial for ensuring the quality and usability of generated datasets in training other models.

Overall, the current state-of-the-art can achieve high-quality hand images of a given pose using diffusion models, however at a high computational cost. Faster approaches such as GANs and VAEs have their own limitations, such as requiring an input image of the target hand pose, and poor generalization to unseen poses and a variety of appearances. In this work, we present an approach that can bridge these gaps, achieving fast, high-quality hand image generation, including previously unseen hand poses.

4.2 Conditional Hand Generation using Latent Space Supervision

In this work we present a novel approach that employs Supervised Random Variable Variational Autoencoder for the synthesis of realistic hand images given a known pose. SRV-VAE facilitates the disentanglement of hand pose and arbitrary appearance vectors, crucial for conditional generation, allowing control over the generation process. An overview of our approach is depicted in Fig. 4.2. By leveraging the stable training and precise encoding capabilities of RV-VAE [102] (Chapter 3), we establish a partially supervised latent space for hand poses by employing conditioning modifications in the VAE architecture (Sec. 4.2.2). This way, hand pose, and appearance features are effectively disentangled within the latent space, producing a visual combination of the two during the forward pass of SRV-VAE (Sec. 4.2.3). The resulting encoder provides an estimation of the input hand pose and encodes the appearance information separately from RGB hand images, while the decoder generates realistic hand images based on specific poses and arbitrary appearance vectors.

4.2.1 Employing RV-VAEs

The formation of the latent space and the generative capabilities of VAEs is attributed to the training procedure of optimizing the ELBO loss as defined by Kingma and Welling in [79] and shown in Eq. (4.1). We follow the standard notation as used in that work [79].

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}) = -D_{KL}(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})]. \tag{4.1}$$

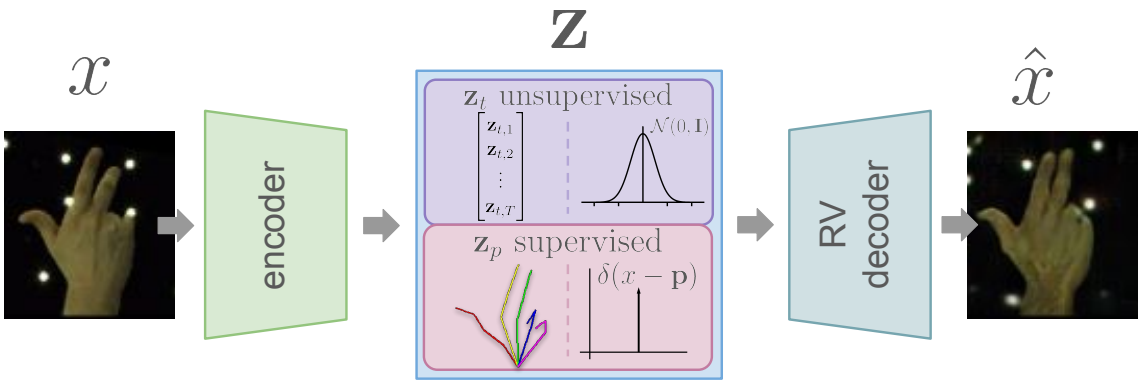


Figure 4.2: The proposed SRV-VAE architecture, for an RGB hand image input x, disentangles the latent space into the unsupervised random variable $\mathbf{z}_t$, and the supervised random variable $\mathbf{z}_p$. The $\mathbf{z}_t$ random variable depicts the encoded texture vector and follows a standard normal distribution, while the $\mathbf{z}_p$ random variable depicts the estimated hand pose and follows a δ distribution. By leveraging the capabilities of the RV-aware architecture we forward these distributions directly to the decoder for reconstructing the input RGB hand image.

The symbols used denote: $q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})$ is the approximate posterior distribution over the latent variable $\mathbf{z}$ given the data $\boldsymbol{x}$, parameterized by φ and typically modeled using a neural network, the encoder. Similarly, $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})$ is the log-likelihood of the data $\boldsymbol{x}$ given the latent variable $\mathbf{z}$, parameterized by θ and typically modeled as another neural network, the decoder. Additionally, $p_{\boldsymbol{\theta}}(\mathbf{z})$ denotes the prior distribution over the latent variable $\mathbf{z}$, often chosen to be a simple distribution like a standard normal distribution $\mathcal{N}(0,1)$. The term $\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})}$ denotes the expectation of the log-likelihood with respect to the approximate posterior distribution over $\mathbf{z}$. Finally, the term D_{KL} denotes the Kullback-Leibler divergence between the two distributions, the approximate posterior $q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})$ and the chosen prior $p_{\boldsymbol{\theta}}(\mathbf{z})$.

Since the second term in this loss formulation, the expectation $\mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})}$ is practically intractable, the authors suggest forming Monte Carlo estimates of it, with the final estimation becoming:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi};\boldsymbol{x}) \simeq \tilde{\mathcal{L}}(\boldsymbol{\theta},\boldsymbol{\varphi};\boldsymbol{x}) = -D_{KL}(q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})\|p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z}^{(l)})), \tag{4.2}$$

where L denotes the number of samples drawn, as implemented by the reparameterization trick.

In our previous chapter (Sec. 3.2) we showed that this sampling can be avoided by using special differentiable RV operations inside the decoder's architecture, and in this way

utilizing the whole distribution of $q_{\boldsymbol{\varphi}}(\mathbf{z}|\boldsymbol{x})$ which describes the encoder's output. This modification has been shown to improve the overall performance of VAEs in terms of reconstruction and image generation without hindering the convergence rate.

Empirical evidence (Sec. 4.3.4) shows that RV-VAEs manage to better disentangle complex data representations, such as those of hands, particularly when using supervised encoded attributes, such as hand poses. This is ascribed to the fact that parts of the latent space (supervised or not) are utilized completely by the network as RVs and prevent any data loss since there are no sampling procedures. This results in the accurate and distinct formulation of encoded regions of the latent space.

4.2.2 Supervised RV-VAE

The beneficial usage of RV-VAE has been reported in Sec. 3.4 for image generation on multiple datasets. However, following this methodology gives no control over any specific attributes we would like to impose in the generation process. Any specific image creation given a requested attribute would require a search in the "opaque" latent space. Therefore we need to enforce some form of structure in the latent space. To achieve this, a natural choice is the language of conditional probability distributions, already used since the original formulation of VAEs. In our case, our goal is to approximate the conditional probability of a hand image given a specific hand pose, disentangled from the rest of the latent space.

The conditioning of image generation is achieved by supervising a subset of the latent space within the RV-VAE training procedure. By incorporating supervision into specific dimensions of the latent space, we aim to impart control over specific aspects of image generation. This novel strategy enables the generation of images conditioned not only on random noise but also on structured latent representations.

Specifically, the whole latent space $S \subseteq \mathbb{R}^n$ (for suitable dimensionality n) is divided into two sub-spaces, the new supervised and the regular unsupervised sub-space. This is achieved by the output of the encoder in the RV-VAE architecture with a modification of the encoder's final layer. Specifically, for a hand pose p with D spatial dimensions and K keypoints, and a latent texture vector of size T, the encoder outputs the parameters of a latent random variable $\mathbf{z} \in \mathbb{R}^{(D\times K+T)\times 2}$ with the last dimension being the two distribution parameters, mean and variance. The random variable $\boldsymbol{z}$ is the concatenation of the random variable $\mathbf{z}_p \in \mathbb{R}^{D\times K}$ and $\mathbf{z}_t \in \mathbb{R}^T$ that depict the hand pose and the encoded latent texture space of the input, respectively.

The general form of the ELBO loss in Eq. (4.1) is modified to incorporate the new conditionality of the latent space, and is given by:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}; \boldsymbol{x}) = -D_{KL}(q_{\boldsymbol{\varphi}}(\mathbf{z}_t|\boldsymbol{x})||p(\mathbf{z}_t)) + \mathbb{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z}_t|\boldsymbol{x}),q_{\boldsymbol{\varphi}}(\mathbf{z}_p|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})] + \mathbb{E}_{p(\mathbf{z}_p|\boldsymbol{x})}[\log q_{\boldsymbol{\varphi}}(\mathbf{z}_p|\boldsymbol{x})]. \quad (4.3)$$

In Eq. (4.3), $q_{\boldsymbol{\varphi}}(\mathbf{z}_t|\boldsymbol{x})$ is the encoder's first output that we want to match to the prior distribution $p(\mathbf{z}_t)$ over the latent variable $\mathbf{z}_t$ (which is the standard normal distribution $\mathcal{N}(0, \boldsymbol{I})$). $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\mathbf{z})$ is the decoder's output, that is, the reconstruction of $\boldsymbol{x}$ given $\mathbf{z}$ where $\mathbf{z}$ is the concatenation of both random variables $\mathbf{z}_t$, $\mathbf{z}_p$. Finally, $q_{\boldsymbol{\varphi}}(\mathbf{z}_p|\boldsymbol{x})$ is the encoder's second output representing the regressed hand pose. This is compared to the true posterior $p(\mathbf{z}_p|\boldsymbol{x})$, that is, the known hand pose $\boldsymbol{x}$.

The encoder outputs the parameterization of a distribution, specifically its first two moments. For a training set of hand images, the ground truth hand poses can be described by a degenerate distribution $P(X = \boldsymbol{p}) = 1$, separately for each hand pose $\boldsymbol{p}$ as they can be considered independent constant random variables with a probability density function described by Dirac delta function $\delta(x - \boldsymbol{p})$. In this paper the goal is hand image generation conditioned on a given hand pose. Therefore, for an input image $\boldsymbol{x}$ with associated hand pose $\boldsymbol{p}$, the optimization of $\mathbb{E}_{p(\mathbf{z}_p|\boldsymbol{x})}[\log q_{\boldsymbol{\varphi}}(\mathbf{z}_p|\boldsymbol{x})]$ term from Eq. (4.3) is equivalent to minimizing the mean $\frac{1}{K}\sum_{i=1}^{K}(\boldsymbol{p}_i - \hat{\boldsymbol{p}}_i)^2$ and the variance $\frac{1}{K}\sum_{i=1}^{K}(\text{var}[\mathbf{z}_{p_i}])^2$ of the Encoder's output, where $\boldsymbol{p}$ and $\hat{\boldsymbol{p}} = \mathbb{E}[\mathbf{z}_p]$ are the ground truth pose with K keypoints and the estimated (by the encoder) hand pose, respectively.

Since the encoder outputs the parameters of distributions, by utilizing the advantages of RV-VAEs we can forward these distributions (of pose and appearance) directly to the decoder. This is possible since the modules inside the decoder are designed to operate on random variables instead of samples (like the reparameterization trick would provide in regular VAEs).

4.2.3 Forward pass of SRV-VAE

Given the formulation described in Sec. 4.2.2, during inference, for a test sample $\boldsymbol{x}$, the encoder will output its estimated hand pose in the form of $\mathbb{E}[\mathbf{z}_p]$.

While the encoder's estimation of hand poses from RGB images might be straightforward, the generation of new images requires further elaboration due to the treatment of the latent data as random variables. Specifically, in order to create an RGB hand image we require a hand pose $\boldsymbol{p}$ that we desire to generate and a random vector texture encodings $\mathbf{z}_t$. By concatenating the flattened vector of $\boldsymbol{p}$ with the texture vector $\mathbf{z}_t$ we create the latent vector $\mathbf{z}$. The decoder then takes this representation of a latent distribution, and in contrast to regular VAE architectures [79], it does not perform a reparameterization trick. Instead, using the RV-VAE approach described in Chapter 3, the two distributions are propagated throughout the layers of the decoder toward the output, with one being the distribution of

encoded hand texture $\mathcal{N}(\mathbb{E}[\mathbf{z}_t], \text{var}[\mathbf{z}_t])$, and the other of the hand pose $\delta(x - \mathbb{E}[\mathbf{z}_p])$. The output will be a generated hand image of $\mathbb{E}[\mathbf{z}_p]$ pose and visualized with the respective texture appearance.

4.3 Experimental Evaluation

We conducted several experiments over a couple of models and datasets to evaluate and assess the performance of the proposed SRV-VAE framework. Specifically, we created variations of a regular VAE architecture [79] and of one based on Soft-Intro-VAE [32], a state-of-the-art approach in generative VAE architectures. These variations were modified to become RV-aware models based on the approach in Sec. 3.3, and further modified to incorporate latent space conditionality, as described in this work. Both model architectures were trained on two datasets, the STB [177] with 2D keypoint hand poses and the InterHand2.6M [97] with 3D keypoint hand poses. Experiments focused on highlighting the quality of hand images conditioned on specific poses (Sec. 4.3.2) as well as the quantitative assessment of the generative capabilities (Sec. 4.3.3). Since this work depends on the beneficial usage of RV-VAEs, we investigated the contribution that RV-aware models provide to the conditionality problem we tackle in this work, compared to regular Supervised Variational Autoencoder (S-VAE) architectures (Sec. 4.3.4). The byproduct of generating images from disentangled latent space can be utilized as appearance transfer between poses (Sec. 4.3.5) or up-sampling sparse hand datasets (Sec. 4.3.6), both strengthen the motivation of this work. All experiments were conducted on the STB [177] and the InterHand2.6M [97] datasets.

4.3.1 Implementation Details

All models and architectures were implemented using the PyTorch [113] library. From the STB dataset, we used sequences labeled as "Random" for training (9k samples) and sequences labeled as "Counting" for testing (9k samples). For hand pose ground truth annotation, we extracted and used 2D hand keypoints. From the InterHand2.6M, we used the train/test split of single right hand images defined by the dataset. From both sequences we removed frames and poses with occlusions or erroneous visual annotations, resulting in 20k training and testing samples. For this dataset, we used 3D hand keypoints.

4.3.2 Qualitative Results

To illustrate the proposed method's generative capabilities we report some qualitative results for all trained methods on both datasets. Specifically, Fig. 4.3 illustrates the generative results on both datasets from SRV-VAE. Furthermore, Figs. 4.4 and 4.5 show the generative results from the Soft-Intro-SRV-VAE architecture. All images were generated by using

(a) STB dataset [177] (b) InterHand2.6M [97]

Figure 4.3: Generated hand images using the SRV-VAE model on the two datasets. Each column has fixed test (unseen) poses, and each row changes the random appearance vector.

unseen test hand poses and by changing their appearance randomly, by concatenating each time a different random texture vector with the test hand pose. We can observe the high quality of generated images resulting from the Soft-Intro-SRV-VAE. This is also attributed to the fact that Soft-Intro-RV-VAE's architecture and training procedure yields a higher quality of images compared to the regular RV-VAE architecture. We can also observe some slight changes in the generated poses (specifically in the last two columns of Fig. 4.4) when iterating through random appearance vectors. This can be attributed to the fact that the specific training set (STB) consists of fewer samples than the InterHand2.6M, making the disentanglement of pose and appearance more challenging.

We also show some generated images using SRV-VAE trained on STB dataset while interpolating through the 2D latent space of appearances (with a fixed pose) Fig. 4.6, and the 2D latent space of appearances and poses Fig. 4.7. We can see how the 2D latent space has captured the complete and diverse set of appearances able to infer each appearance sample to any pose.

4.3.3 Quantitative Results

A commonly used and reliable method to quantitatively assess the quality of generated samples of a generative model based on a training dataset is by using the FID [32, 102, 111]. As explained, this metric measures the distance between the distributions of real and generated images, providing a quantitative assessment of how similar the generated images are to the real ones.

Given the nature of the problem this work aims to tackle, the generative process does

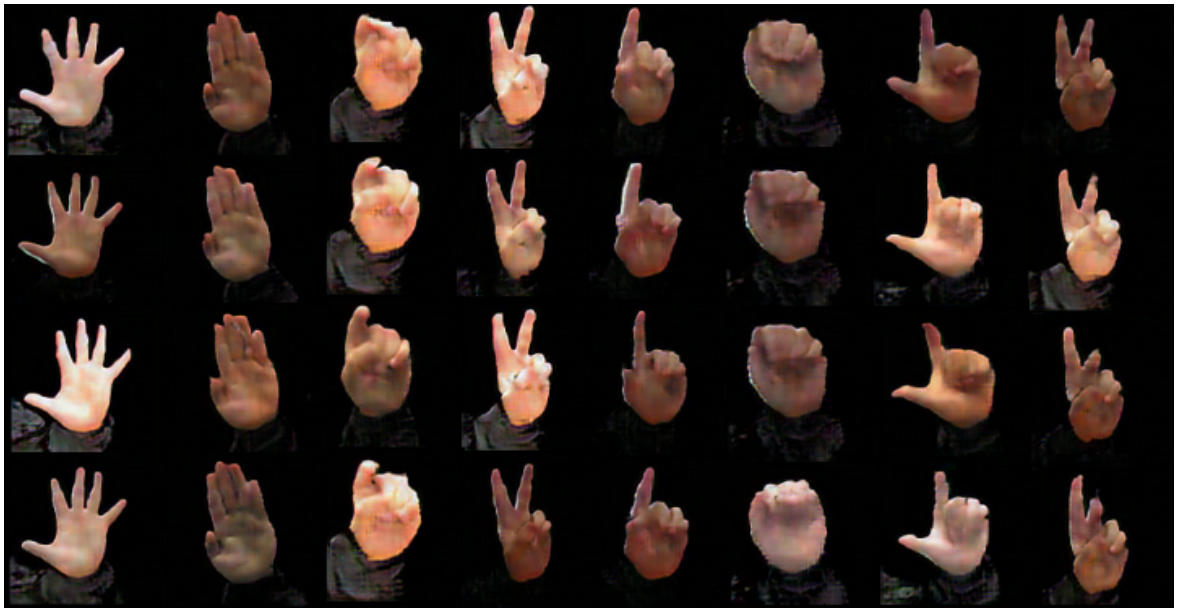


Figure 4.4: Generated hand images using the Soft-Intro-SRV-VAE model on STB dataset [177]. Each column has fixed test (unseen) poses, and each row changes the random appearance vector.

not depend only on the training set but also on a test set, as generated images are conditioned on the unseen poses of that test set. Therefore, the comparison between the distributions of training and generated images is not indicative, since by design, we want them to differ. For that reason, we need to consider two different comparisons: (a) between the distributions of training images and generated images conditioned on poses seen in the training set, and (b) between the distributions of test images and generated images conditioned on unseen poses from the test set. We want those differences not to be far apart, showing that the quality does not change drastically when generated from different distributions.

In Tab. 4.1 we report the FIDs that measure the distance between the distribution of training set images and the distribution of generated images from poses of the same training set. Respectively, in Tab. 4.2 we present the FIDs that measure the distance between the distribution of test set images and the distribution of generated images from unseen poses of the test set. As expected, differences when comparing the generated images from seen and unseen poses are very small, indicating a good quality of hand images.

Figure 4.5: Generated hand images using the Soft-Intro-SRV-VAE model on InterHand2.6M dataset [97]. Each column has fixed test (unseen) poses, and each row changes the random appearance vector.

Table 4.1: FID values that measure distances for generated images between real training set distribution and generated (known training poses) distribution for different combinations of datasets and methods.

Method	Dataset	**FID ↓**
SRV-VAE	STB	25.27
SRV-VAE	InterHand2.6M	16.30
Soft-Intro-SRV-VAE	STB	11.07
Soft-Intro-SRV-VAE	InterHand2.6M	10.59

4.3.4 SRV-VAE vs regular S-VAE

In this section we want to emphasize the importance of using the proposed RV formulation that was presented in Chapter 3, for the task we stated in this chapter of conditional hand image generation. To do this, we are required to create regular Supervised Variational Autoencoder models that try to disentangle the appearance and the pose in the latent space the same way SRV-VAEs do, in order to generate new hand images conditioned on poses. The S-VAE models must encode the input hand image into two distributions, one for the appearance (Nomral) and one for the pose (Dirac). To evaluate the significance of RV-aware models we need to compare the generative capabilities between the

Figure 4.6: Generated hand images using the SRV-VAE model on STB dataset [177] through interpolating the 2D latent appearance space with a fixed pose.

Table 4.2: FID values that measure distances for generated images between real test set distribution and generated (unseen test poses) distribution for different combinations of datasets and methods.

Method	Dataset	**FID** ↓
SRV-VAE	STB	26.84
SRV-VAE	InterHand2.6M	16.13
Soft-Intro-SRV-VAE	STB	14.62
Soft-Intro-SRV-VAE	InterHand2.6M	9.27

SRV-VAE and S-VAE models and the accuracy of the disentanglement. For the former, we can compare the FID values between the models in question, while for the latter we can evaluate the encoded poses that the models learn to disentangle from the appearances.

To make a fair comparison between SRV-VAE and regular S-VAE that specialize in conditioning the latent space as stated for the generative purposes, we are required to sample

Figure 4.7: Generated hand images using the SRV-VAE model on STB dataset [177] through interpolating the 2D latent appearance space and the test pose set.

from two sub-spaces (the pose and appearance) for the regular S-VAE. This is crucial for the training of the S-VAE network, while the SRV-VAE takes all distributions as they are (Sec. 4.2.2).

We created and trained two networks, the regular S-VAE and Soft-Intro-S-VAE, on the STB [177] and InterHand2.6M [97] datasets. We then measured the FIDs as described in Sec. 4.3.3 for all models and datasets in the cases of generating hand images from training set poses and unseen test poses. Table 4.3 reports the FIDs on the regular S-VAE models compared to the previously reported FIDs of the RV variants. We observe that the RV modules in the networks contribute significantly towards the generative capabilities of the method. This improvement can be attributed to the fact that SRV-VAE does not depend on any sampling during training, whereas the regular S-VAE requires sampling from two spaces. This double sampling introduces even more uncertainty into the training pipeline.

To understand which model handles better the disentanglement between the appearance and the pose, we can treat them as pose estimators. This way we can measure the

Table 4.3: FID values of generated images from all non-RV models trained on all datasets, compared to their RV-aware counterpart models.

	InterHand2.6M [97]		STB [177]	
Models	train	test	train	test
S-VAE	17.19	16.96	26.41	29.72
SRV-VAE	**16.3**	**16.13**	**25.27**	**26.84**
Soft-Intro-S-VAE	11.43	9.95	11.48	14.77
Soft-Intro-SRV-VAE	**10.59**	**9.27**	**11.07**	**14.62**

Table 4.4: MPJPE of encoded (estimated) poses from all non-RV models trained on all datasets, compared to their RV-aware counterpart models.

Models	InterHand2.6M [97]	STB [177]
S-VAE	112.49	9.39
SRV-VAE	**112.27**	**5.96**
Soft-Intro-S-VAE	84.23	15.35
Soft-Intro-SRV-VAE	**83.01**	**14.97**

Euclidean distance between the encoded pose and the ground truth pose that the input image has, since the closer the pose is to the ground truth the better the disentanglement. For that end, we report in Tab. 4.4 the Mean per Joint Positional Error (MPJPE) values in pixel space for the STB dataset, and in *mm* for the InterHand2.6M dataset. As observed, the RV-aware models manage to have poses closer to the ground truth in all cases. This reveals to us that even though the majority of modifications are concentrated within the decoder (RV modules), they are beneficial to the entire network pipeline.

4.3.5 Appearance Transfer

The formulation we use in this work disentangles the appearance and pose of an image depicting a hand as explained in Sec. 4.2.2. The disentanglement is enforced by the encoder of the network, which outputs separately: an estimation of the encoded texture, random variable $\boldsymbol{z}_t$, and an estimation of the hand pose of the input, random variable $\boldsymbol{z}_p$. This implies that a trained encoder can yield an appearance estimation of an input image that can be "enforced" on a different pose to generate new images as seen in Fig. 4.8.

4.3.6 Quantitative Evaluation on Downstream Problems

Given the nature of the proposed approach, a meaningful question to ask is whether it can be used to improve the quality of existing datasets, by performing a domain-aware data

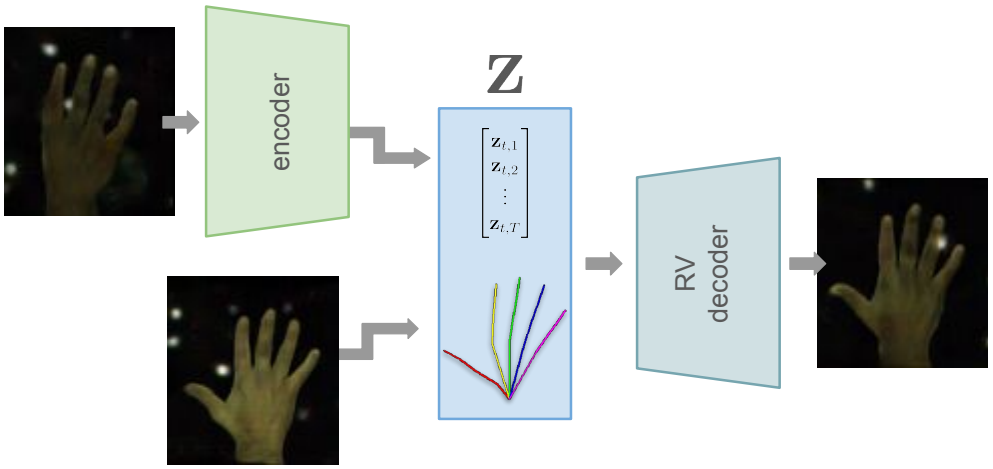


Figure 4.8: The trained encoder can be used to extract the appearance of an input image that can be transferred to a different pose via the decoder.

Table 4.5: MPJPE comparing the performance of the implemented keypoint estimator on original and augmented versions of the two datasets we experimented on.

Dataset	Original	Augmented
STB (pixel space)	11.74	**10.59**
InterHand2.6M (mm)	**11.51**	11.73

augmentation. Specifically, given a hand pose dataset, one can use the proposed approach to generate more training samples, by combining hand poses from one subject with the appearance of another, effectively "upsampling" the existing dataset into a denser one. Such an approach would enrich the existing dataset, potentially improving the accuracy of a keypoint detector trained on the resulting augmented data. To assess this hypothesis, we implemented a hand keypoint detector network based on the ViT architecture [36]. Specifically, using ViT as the backbone, we added a head for the prediction of the hand joint positions, inspired by [114]. The resulting network consists of 10M parameters.

We trained this detector on two subsets of the datasets we used, STB [177] and InterHand2.6M [97]. Each was augmented, and we compared the performance with and without the augmented data. Specifically, we used subsets of the original InterHand2.6M and STB datasets consisting of 4000 and 1800 images, respectively. These datasets were augmented with 8000 and 3600 images. We also selected a test set for each, consisting of 4100 and 1800 images, respectively. To ensure a fair comparison, the networks trained on the original datasets were trained for twice as many epochs as those with augmented datasets,

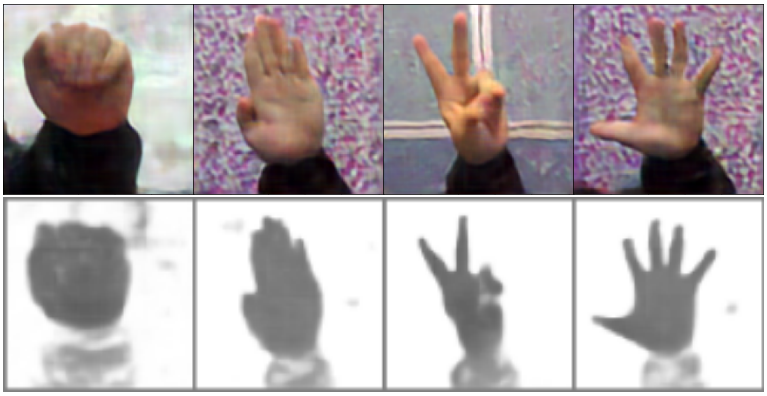


Figure 4.9: Depth estimation results using the model from Sec. 2.2 on the synthetic hand images created with SRV-VAE.

matching the doubled amount of training data in the latter.

As shown in Tab. 4.5 by the MPJPE, the augmentation is always at least non-disrupting, and in the case of one of the datasets, it helps significantly reduce the resulting estimation error (by 9.79%). This is attributed to the diversity difference between the two datasets: STB has only one subject gesturing a limited range of hand gestures, whereas InterHand2.6M is more diverse, both in subjects, and in the performed gestures. These results show that our approach is particularly beneficial as a dataset augmentation tool for the case of small or non-diverse datasets, while not hurting the final performance in other cases.

4.3.7 Depth Estimation of Synthetic Hand Images

For the general problem stated in this thesis, of multi-domain hand image synthesis, we attempted to close the loop of our implemented systems. Having our pre-trained model from Chapter 2 on the STB train set, we employed it on our generated hand images.

Specifically, we generated hand images with our SRV-VAE model by "skinning" the STB test set poses with random appearances. Then we estimated the depth information of those generated images with our stacked hourglass model from Sec. 2.2. We compared the estimated depth images, with the depth images of the corresponding poses used to generate the synthetic images. This resulted in an average depth error of $E = 38.79mm$ and $IoU = 0.69$. In Fig. 4.9 we visualize some results of the synthetic RGB input and the estimated depths. As observed and stated in Sec. 4.3.2 there are some slight deviations in the synthetic images from the original pose, these deviations also affect the depth comparison without implying an incorrect depth estimation.

4.4 Summary

In this chapter, we introduced SRV-VAE models, a novel supervised variational autoencoder framework designed to generate realistic hand images conditioned on specific hand poses. Our approach effectively addresses the challenges associated with hand image generation, particularly the need for precise control over pose and the synthesis of anatomically accurate hand images. By leveraging the strengths of the RV-VAE architecture and incorporating supervision into the latent space, SRV-VAE enables the disentanglement of pose and appearance, resulting in high-quality and diverse hand image outputs.

The improvements observed through our experiments, in both qualitative and quantitative metrics, underscore the potential of SRV-VAE as a capable generative model for synthetic data creation in the domain of hand image synthesis. SRV-VAE represents a significant step forward in the field of hand image generation, offering a robust and flexible tool for creating synthetic hand images with precise pose control. This outcome can be beneficial even in augmenting limited datasets with the task of enhancing the performance of hand pose estimation models, as our experiments showed. Moreover, our results have strengthened our claims from the previous chapter (Chapter 3) about the beneficial usage of RV-aware modules.

The SRV-VAE group of models completes our endeavor of *advanced hand image synthesis in multi-domain*. Being deployed in the RGB domain, we showed that we are capable of generating new hand RGB images that can be used even as input in our first, hand depth estimation, system in Chapter 2. While poses extracted from the depth information of hands can be used to be skinned with different appearances. This joint utilization facilitates the synthesis and augmentation of larger hand datasets in multi-domains, contributing to the ever-growing demand for data.

Chapter 5
Conclusions

5.1 Synopsis of Contributions

In the era of Deep Neural Networks, computer vision has reached a point of huge demand for data, either for training or evaluation purposes. Synthetic generation of data is a viable solution that is not widely used in the field of hand-related research. Due to the numerous tasks that span multiple domains and, specifically, on RGB and depth, synthetic approaches that generate hand images in those two domains are very promising. For that reason, we require innovative methodologies that achieve depth estimation, efficient learning, and controllable data generation for the task of hand image synthesis. Toward this end, in this dissertation we presented solutions to achieve the aforementioned objectives.

Firstly, we presented a novel depth estimation method capable of reconstructing the depth information of human hands from monocular RGB images. By leveraging a specially designed lightweight CNN and incorporating intermediate supervision using hand segmentation masks, we demonstrated a significant step toward bridging the gap between RGB and RGBD modalities. To train and evaluate our model, we compiled a publicly available dataset, named HandRGBD, that contains 20,601 aligned hand images of RGB and depth. Our experimental evaluations showed the robustness of depth estimation on test sets, and proved the utilization of our model on hands-on problems.

Before introducing generative solutions to our problem, we managed to improve already existing tools to enable efficiency and more accuracy for our tasks. Specifically, we enhanced the generative family of models, Variational Autoencoders, by introducing RV-aware modifications to the decoder activations as continuous random variables. This innovation enables structured latent spaces with improved reconstruction quality and generative fidelity. By allowing conditional traversal of the latent space, RV-VAEs provide precise control over the generative process while maintaining computational efficiency. These models demonstrate capabilities comparable to more complex architectures, like Generative Adversarial Networks or Transformer-based methods, making them particularly suitable for tasks like advanced hand image synthesis in the RGB domain.

Building on these advancements, we finally introduced Supervised Random Variable Variational Autoencoder, a supervised generative framework that disentangles pose and appearance in latent space for photorealistic controllable hand image synthesis. This approach ensures accurate outputs and precise control over pose conditioning, addressing critical challenges in hand image generation. Additionally, SRV-VAE supports dataset augmentation by generating diverse and high-quality synthetic data, which can enhance the performance of downstream hand pose estimation models. The integration of RV-aware modules further amplifies its efficacy experimentally.

Together, these contributions form a cohesive pipeline that connects depth estimation, structured generative modeling, and controllable image creation into a unified framework targeted at hand image synthesis. The ability to utilize synthetic RGB images as inputs for depth estimation or to synthesize RGB images from poses derived from depth exemplifies the synergy between these components. This joint framework has the potential to meet the growing demand for high-quality hand data while addressing challenges in efficiency, and realism, laying the foundation for future advancements in generative systems and hand analysis.

5.2 Directions for Future Work and Research

Future work can occur from the collective objective of this dissertation, as well as from the individual tasks we managed to tackle.

Specifically, viewing the hand image synthesis as a whole, a possible direction is to include more useful modalities that target additional domains. Surface normals, hand part segmentation, and even optical flow for video inputs are some examples of such domains that can enhance the complete pipeline. They can be addressed as separate tasks, or be integrated in the task of hand depth estimation with the cost of increasing the size of the model.

Further future plans, for the hand depth estimation method, include refining the results using higher-accuracy data acquired by a laser-based depth sensor. In addition, an interesting direction is testing our method for other depth reconstruction tasks, as already presented in the Appendix A for the task of bathymetry depth estimation. Further improvements could also include the integration of attention mechanisms for better intermediate supervision of the hand's mask.

Regarding the integration of Random Variables in ANNs, future work could focus on implementing more ANN layers and activation functions to become RV-aware in order to extend the proposed mathematical toolbox. Furthermore, future research could be targeted towards investigating the applicability of our approach to other network types, beyond VAEs, such as diffusion models where each noising/de-noising step could be achieved deterministically with our framework. Some preliminary experiments for employment of RV

modules in training augmented data that are reported in Appendix B justify this.

Finally, for the task of conditional hand image generation, future directions include assessing the performance of SRV-VAE with pose estimation techniques, to further explore the quality and accuracy of the generated hand images, and investigate its correlation with achieving better disentanglement. Additionally, an interesting development would be to expand the SRV-VAE to disentangle the latent space even further, specifically the appearance domain into separate subdomains (shape, texture, illumination, etc.), and investigate the use of SRV-VAE in other aspects of real-world data, beyond hands (e.g. face image synthesis conditioned to an emotion).

Bibliography

[1] Kinect for XBox One. https://en.wikipedia.org/wiki/Kinect. [Online; Accessed 11-Nov-2024].

[2] Microsoft HoloLens. https://www.microsoft.com/en-CY/hololens. [Online; Accessed 11-Nov-2024].

[3] Midjourney. https://www.midjourney.com. [Online; Accessed 11-Nov-2024].

[4] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[5] Diar Abdlkarim, Massimiliano Di Luca, Poppy Aves, Mohamed Maaroufi, Sang-Hoon Yeo, R Chris Miall, Peter Holland, and Joeseph M Galea. A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the meta quest 2. *Behavior research methods*, 56(2):1052–1063, 2024.

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[7] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan L Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22861–22872, 2024.

[8] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. *ICCV*, 2015 Inter:1949–1957, 2015.

[9] Patrizia Baraldi, Enrico De Micheli, and Sergio Uras. Motion and depth from optical flow. In *Alvey Vision Conference*, pages 1–4, 1989.

[10] H G Barrow and J M Tenenbaum. Interpreting Line Drawings as Three-Dimensional Surfaces. *Artificial Intelligence*, 17(3), 1981.

[11] Rilwan R Basaru, Gregory G Slabaugh, Christopher Child, and Eduardo Alonso. HandyDepth : Example-based Stereoscopic Hand Depth Estimation using Eigen Leaf Node Features. In *IWSSIP*, pages 1–4, 2016.

[12] Rilwan Remilekun Basaru, Chris Child, Eduardo Alonso, and Gregory Slabaugh. Data-driven Recovery of Hand Depth using Conditional Regressive Random Forest on Stereo Images. *IET Computer Vision*, 2018.

[13] Felix Berkhahn, Richard Keys, Wajih Ouertani, Nikhil Shetty, and Dominik Geißler. Augmenting variational autoencoders with sparse labels: A unified framework for unsupervised, semi-(un) supervised, and supervised learning. *arXiv preprint arXiv:1908.03015*, 2019.

[14] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

[15] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, June 2023.

[16] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[17] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

[18] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[19] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*, pages 1021–1030. IEEE, 2017.

[20] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[21] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2019.

[22] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[23] Xinghao Chen, Guijin Wang, Hengkai Guo, and Cairong Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 2018.

[24] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating Depth from RGB and Sparse Sensing. *arXiv preprint arXiv:1804.02771*, pages 1–20, 2018.

[25] Zhaoxi Chen, Gyeongsik Moon, Kaiwen Guo, Chen Cao, Stanislav Pidhorskyi, Tomas Simon, Rohan Joshi, Yuan Dong, Yichen Xu, Bernardo Pires, He Wen, Lucas Evans, Bo Peng, Julia Buffalini, Autumn Trimble, Kevyn McPhail, Melissa Schoeller, Shoou-I Yu, Javier Romero, Michael Zollhöfer, Yaser Sheikh, Ziwei Liu, and Shunsuke Saito. URhand: Universal relightable hands. In *CVPR*, 2024.

[26] Jian Cheng, Yanguang Wan, Dexin Zuo, Cuixia Ma, Jian Gu, Ping Tan, Hongan Wang, Xiaoming Deng, and Yinda Zhang. Efficient virtual view selection for 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 419–426, 2022.

[27] Wencan Cheng, Eunji Kim, and Jong Hwan Ko. Handdagt: A denoising adaptive graph transformer for 3d hand pose estimation. In *European Conference on Computer Vision*, pages 35–52. Springer, 2025.

[28] Jimei Yang Bryan Russel Max Argus Christian Zimmermann, Duygu Ceylan and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[29] Florian Ciurea, Kartik Venkataraman, Gabriel Molina, and Dan Lelescu. System and methods for measuring depth using an array camera employing a bayer filter, September 1 2015. US Patent 9,123,118.

[30] Nadav Cohen, Or Sharir, Yoav Levine, Ronen Tamari, David Yakira, and Amnon Shashua. Analysis and design of convolutional networks via hierarchical tensor decompositions. *arXiv preprint arXiv:1705.02302*, 2017.

[31] J Cook. The mean-field theory of a q-state neural network model. *Journal of Physics A: Mathematical and General*, 22(12):2057, 1989.

[32] Tal Daniel and Aviv Tamar. Soft-introvae: Analyzing and improving the introspective variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4391–4400, June 2021.

[33] Erick Delage, Honglak Lee, and Andrew Y Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In *Robotics Research*, pages 305–321. Springer, 2007.

[34] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

[35] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[37] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[38] Philippe Dreuw, Thomas Deselaers, Daniel Keysers, and Hermann Ney. Modeling image variability in appearance-based gesture recognition. *Proc. of the ECCV 2006 3rd Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP), 12 May, Graz, Austria*, pages 7–18, 2006.

[39] H Pallab Jyoti Dutta, Manas Kamal Bhuyan, Debanga Raj Neog, Karl Fredric MacDorman, and Rabul Hussain Laskar. Efficient hand segmentation for rehabilitation tasks using a convolution neural network with attention. *Expert Systems with Applications*, 234:121046, 2023.

[40] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2650–2658, 2015.

[41] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *NIPS*, pages 1–9, 2014.

[42] Lingzhu Xiang et al. libfreenect2: Release 0.2, April 2016. `https://github.com/OpenKinect/libfreenect2`.

[43] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-convs: camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11826–11835, 2019.

[44] Brian Fehrman and Jeff McGough. Depth mapping using a low-cost camera array. In *2014 Southwest symposium on image analysis and interpretation*, pages 101–104. IEEE, 2014.

[45] Alexandre RJ François and Gérard G Medioni. Interactive 3d model extraction from a single image. *Image and Vision Computing*, 19(6):317–328, 2001.

[46] Mathieu Garon, Pierre-Olivier Boulet, Jean-Philippe Doironz, Luc Beaulieu, and Jean-François Lalonde. Real-time high resolution 3d data on the hololens. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pages 189–191. IEEE, 2016.

[47] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 10833–10842, 2019.

[48] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey. *arXiv preprint arXiv:2101.00734*, 2021.

[49] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019.

[50] Charlie Giattino, Edouard Mathieu, Veronika Samborska, and Max Roser. Artificial intelligence. *Our World in Data*, 2023. https://ourworldindata.org/artificial-intelligence.

[51] Francisco Gomez-donoso, Sergio Orts-escolano, and Miguel Cazorla. Large-scale Multiview 3D Hand Pose Dataset. pages 1–23.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[53] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12, 2009.

[54] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7978–7987, 2020.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[56] Lei He, Guanghui Wang, and Zhanyi Hu. Learning Depth from Single Images with Deep Neural Network Embedding Focal Length. *IEEE Transactions on Image Processing*, (April), 2018.

[57] Ari Heljakka, Arno Solin, and Juho Kannala. Towards photographic image manipulation with balanced growing of generative autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3120–3129, 2020.

[58] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. HoloDesk: Direct 3D Interactions with a Situated See-Through Display. *CHI'12*, page 2421, 2012.

[59] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[60] Berthold K P Horn and Michael J Brooks. The Variational Approach to Shape from Shading. *Computer Vision, Graphics, and Image Processing*, 208:174–208, 1986.

[61] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[62] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *CVPR*, 2017-Janua, 2017.

[63] Huaibo Huang, Zhihang Li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Information Processing Systems*, 2018.

[64] Intel. Intel RealSense LiDAR camera l515 2020. `https://www.intelrealsense.com/lidar-camera-l515`, 2020. [Online; Accessed 11-Nov-2024].

[65] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[66] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[67] Aaron S. Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. *ICCV*, 2017-Octob:1031–1039, 2017.

[68] Priyank Jaini, Pascal Poupart, and Yaoliang Yu. Deep homogeneous mixture models: representation, separation, and approximation. *Advances in Neural Information Processing Systems*, 31, 2018.

[69] Changsoo Je, Sang Wook Lee, and Rae-Hong Park. High-contrast color-stripe pattern for rapid structured-light range imaging. In *European Conference on Computer Vision*, pages 95–107. Springer, 2004.

[70] Dimitris Kalatzis, David Eklund, Georgios Arvanitidis, and Søren Hauberg. Variational autoencoders with riemannian brownian motion priors. *arXiv preprint arXiv:2002.05227*, 2020.

[71] Vivek Kanhangad, Ajay Kumar, and David Zhang. Contactless and pose invariant biometric identification using hand surface. *IEEE TIP*, 20(5):1415–1424, 2011.

[72] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[73] Yuta Kawachi, Yuma Koizumi, and Noboru Harada. Complementary set variational autoencoder for supervised anomaly detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2366–2370. IEEE, 2018.

[74] James M. Keller, Richard M. Crownover, and Robert Y U Chen. Characteristics of Natural Scenes Related to the Fractal Dimension. *IEEE Trans. on PAMI*, (5):621–627, 1987.

[75] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 119–137, 2013.

[76] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. *arXiv preprint arXiv:2408.12569*, 2024.

[77] Hyeongwoo Kim, Christian Richardt, and Christian Theobalt. Video depth-from-defocus. In *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 370–379, 2016.

[78] Minyoung Kim. Gaussian process modeling of approximate inference errors for variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2022.

[79] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[80] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *... Science Department, University of Toronto, Tech. ...*, 2009.

[81] Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems*, 31, 2018.

[82] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[83] 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.

[84] Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2019.

[85] Jeongho Lee, Jaeyun Kim, Seon Ho Kim, and Sang-Il Choi. Enhancing 3d hand pose estimation using shaf: synthetic hand dataset including a forearm. *Applied Intelligence*, pages 1–14, 2024.

[86] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[87] Lijun Li, Linrui Tian, Xindi Zhang, Qi Wang, Bang Zhang, Liefeng Bo, Mengyuan Liu, and Chen Chen. Renderih: A large-scale synthetic dataset for 3d interacting hand pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20395–20405, 2023.

[88] Stephen Lin, Yuanzhen Li, Sing Bing Kang, Xin Tong, and Heung-Yeung Shum. Diffuse-specular separation and depth recovery from image sequences. In *European conference on computer vision*, pages 210–224. Springer, 2002.

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

[90] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[91] David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[92] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In *CVPR 2018*, pages 5667–5675, 2018.

[93] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[94] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016.

[95] Giulio Marin, Fabio Dominio, and Pietro Zanuttigh. Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimedia Tools and Applications*, 75(22):14991–15015, 2016.

[96] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[97] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020.

[98] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[99] Takayuki Nagai, Masaaki Ikehara, and Akira Kurematsu. Hmm-based surface reconstruction from single images. *Systems and Computers in Japan*, 38(11):80–89, 2007.

[100] Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. Handiffuser: Text-to-image generation with realistic hand appearances. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[101] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 2016.

[102] Vassilis C Nicodemou, Iason Oikonomidis, and Antonis Argyros. Rv-vae: Integrating random variable algebra into variational autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 196–205, 2023.

[103] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics*, 38(6):184:1–184:15, 2019.

[104] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[105] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE international conference on computer vision Workshops*, pages 585–594, 2017.

[106] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Generalized Feedback Loop for Joint Hand-Object Pose Estimation. mar 2019.

[107] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*, pages 101.1–101.11, Dundee, UK, 2011.

[108] Sergiu Oprea, Giorgos Karvounas, Pablo Martinez-Gonzalez, Nikolaos Kyriazis, Sergio Orts-Escolano, Iason Oikonomidis, Alberto Garcia-Garcia, Aggeliki Tsoli, Jose Garcia-Rodriguez, and Antonis Argyros. H-gan: the power of gans in your hands. In *2021 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2021.

[109] Kemal Egemen Ozden, Konrad Schindler, and Luc Van Gool. Simultaneous segmentation and 3d reconstruction of monocular image sequences. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[110] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445. IEEE, 2018.

[111] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 823–832, June 2021.

[112] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[113] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[114] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9826–9836, 2024.

[115] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *CVPR 2018*, 2018.

[116] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14104–14113, 2020.

[117] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

[118] Rolandos Alexandros Potamias, Stylianos Ploumpis, Stylianos Moschoglou, Vasileios Triantafyllou, and Stefanos Zafeiriou. Handy: Towards a high fidelity

3d hand shape and appearance model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4670–4680, June 2023.

[119] Ammar Qammaz, Nikolaos Vasilikopoulos, Iason Oikonomidis, and Antonis A Argyros. Y-map-net: Real-time depth, normals, segmentation, multi-label captioning and 2d human pose in rgb images. *arXiv preprint arXiv:2411.10334*, 2024.

[120] Neng Qian, Jiayi Wang, Franziska Mueller, Florian Bernard, Vladislav Golyanik, and Christian Theobalt. HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

[121] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[122] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[123] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*, 2015.

[124] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019.

[125] Zhou Ren, Jingjing Meng, and Junsong Yuan. Depth camera based hand gesture recognition and its applications in human-computer-interaction. In *2011 8th International Conference on Information, Communications & Signal Processing*, pages 1–5. IEEE, 2011.

[126] Mohammad Rezaei, Razieh Rastgoo, and Vassilis Athitsos. Trihorn-net: a model for accurate depth-based 3d hand pose estimation. *Expert Systems with Applications*, 223:119922, 2023.

[127] Grégory Rogez, Maryam Khademi, James S. Supancic, J.M.M. Montiel, and Deva Ramanan. 3D Hand Pose Detection in Egocentric RGB-D Images. In *ECCVW*, 2014.

[128] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[129] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[130] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[131] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A Kakadiaris. 3D Human pose estimation : A review of the literature and analysis of covariates. *CVIU*, 152:1–20, 2016.

[132] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning Depth from Single Monocular Images. In *NIPS*, 2006.

[133] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: learning 3D scene structure from a single still image. *2007 IEEE 11th International Conference on Computer Vision*, 31(5):824–840, 2007.

[134] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3D : Depth Perception from a Single Still Image. *Aaai*, pages 1571–1576, 2008.

[135] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[136] Yujia Shen, Arthur Choi, and Adnan Darwiche. Tractable operations for arithmetic circuits of probabilistic models. *Advances in Neural Information Processing Systems*, 29, 2016.

[137] Ilan Shimshoni, Yael Moses, and Michael Lindenbaum. Shape Reconstruction of 3D Bilaterally Symmetric Surfaces. *International Journal of Computer Vision*, 15, 2000.

[138] N. Siddharth, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Noah D. Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5927–5937. Curran Associates, Inc., 2017.

[139] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. *CVPR*, 2017-Janua:4645–4653, 2017.

[140] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[141] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 130(3):1820–1852, 2020.

[142] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[143] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29:3738–3746, 2016.

[144] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[145] Melvin Dali Springer. The algebra of random variables. Technical report, 1979.

[146] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, pages 2456–2463, 2013.

[147] A.K Subramanian. Pytorch-vae. `https://github.com/AntixK/PyTorch-VAE`, 2020.

[148] Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*, 2019.

[149] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *CVPR*, pages 824–832, 2015.

[150] James S Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE international conference on computer vision*, pages 1868–1876, 2015.

[151] James Steven Supančič, Gregory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *International Journal of Computer Vision*, 126(11):1180–1198, 2018.

[152] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*, pages 3786–3793, 2014.

[153] Michael W. Tao, Jong Chyi Su, Ting Chun Wang, Jitendra Malik, and Ravi Ramamoorthi. Depth Estimation and Specular Removal for Glossy Surfaces Using Point and Line Consistency with Light-Field Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1155–1169, 2016.

[154] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM ToG*, 35(6):1–11, 2016.

[155] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.

[156] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *SIGGRAPH*, 33(5):1–10, 2014.

[157] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Trans. on PAMI*, 24(9):1226–1238, 2002.

[158] Dustin Tran, Alp Kucukelbir, Adji B Dieng, Maja Rudolph, Dawen Liang, and David M Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

[159] Aggeliki Tsoli, Antonis Argyros, et al. Patch-based reconstruction of a textureless deformable 3d surface from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[160] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation. *IJCV*, 118(2):172–193, 2016.

[161] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[162] Aydin Varol, Mathieu Salzmann, Engin Tola, and Pascal Fua. Template-Free Monocular Reconstruction of Deformable Surfaces. In *ICCV*, 2009.

[163] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric Inference of 3D Human Body Shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–27, 2018.

[164] Nikolaos Vasilikopoulos, Drosakis Drosakis, and Antonis Argyros. D-pose: Depth as an intermediate representation for 3d human pose and shape estimation. *arXiv preprint arXiv:2410.04889*, 2024.

[165] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. *Advances in Neural Information Processing Systems*, 34:13189–13201, 2021.

[166] Robin Willink. *Measurement uncertainty and probability*. Cambridge University Press, 2013.

[167] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021.

[168] Chi Xu and Li Cheng. Efficient Hand Pose Estimation from a Single Depth Image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2013.

[169] Hongteng Xu, Dixin Luo, Ricardo Henao, Svati Shah, and Lawrence Carin. Learning autoencoders with relational regularization. In *International Conference on Machine Learning*, pages 10576–10586. PMLR, 2020.

[170] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[171] Na-Eun Yang, Yong-Gon Kim, and Rae-Hong Park. Depth hole filling using the depth distribution of neighboring regions of depth holes in the kinect sensor. In *2012 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2012)*, pages 658–661. IEEE, 2012.

[172] Yue Yang, Atith N Gandhi, and Greg Turk. Annotated hands for generative models. *arXiv preprint arXiv:2401.15075*, 2024.

[173] Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A survey on human motion analysis from depth data. In *Time-of-flight and depth imaging. sensors, algorithms, and applications*, pages 149–187. Springer, 2013.

[174] Shanxin Yuan, Guillermo Garcia-Hernando, Bjorn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, and Tae-Kyun Kim. Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals. In *CVPR 2018*, 2018.

[175] Shanxin Yuan and Tae-kyun Kim. BigHand2 2M Benchmark: Hand Pose Dataset and State of the Art Analysis. In *CVPR*, pages 15–20, 2017.

[176] Kevan Yuen. VIVA Hand Tracking Challenge, 2015.

[177] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3D Hand Pose Tracking and Estimation Using Stereo Matching. *arXiv:1610.07214*, 2016.

[178] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[179] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[180] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[181] Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless C Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3. Citeseer, 2012.

[182] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

Appendix A
Publications, Models and Datasets

Publications

The research activity related to this thesis has so far produced the following publications (ordered by publication date):

(1) Nicodemou, V.C., Oikonomidis, I., Tzimiropoulos, G. and Argyros, A., 2020, July. Learning to infer the depth map of a hand from its color image. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

(2) Alevizos, E., Nicodemou, V.C., Makris, A., Oikonomidis, I., Roussos, A. and Alexakis, D.D., 2022. Integration of Photogrammetric and Spectral Techniques for Advanced Drone-Based Bathymetry Retrieval Using a Deep Learning Approach. *Remote Sensing, 14*(17), p.4160.

(3) Nicodemou, V.C., Oikonomidis, I. and Argyros, A., 2023. RV-VAE: Integrating Random Variable Algebra into Variational Autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 196-205).

(4) Nicodemou, V.C., Oikonomidis, I., Karvounas, G. and Argyros, A., 2024. Conditional Hand Image Generation using Latent Space Supervision in Random Variable Variational Autoencoders. In *IEEE/CVF European Conference on Computer Vision Workshops (HANDS 2024 - ECCVW 2024)*, IEEE

(5) Makris, A., Nicodemou, V.C., Alevizos, E., Oikonomidis, I., Alexakis, D.D. and Roussos, A., 2024. Refraction-Aware Structure from Motion for Airborne Bathymetry. *Remote Sensing, 16*(22), p.4253.

In more details (and with regard to the contributions of this thesis), (1) corresponds to the work (model, experiments, and results) of hand depth estimation from single RGB image as reported in Chapter 2. Publications (2) and (5) present an extension to the model formulated in Chapter 2 applied to the task of depth estimation for the purposes of shallow bathymetry mapping with (as presented in (5)) and without (as presented in (2)) refraction-aware integration. Publication (3) contains the research of Chapter 3 along with the defined RV modules and RV-aware modified networks together with the complete experi-

mental evaluation. Finally, (4) is the publication regarding the majority of work presented in Chapter 4, as further experiments and investigations were presented in this thesis.

Models and Datasets

In the context of this thesis, the following systems and models were developed and evaluated:

- The hand depth estimation CNN model. With its extension for depth estimation of the seafloor for shallow bathymetry mapping.
- The *HandRGBD* dataset of aligned RGB and depth hand images.
- The design and open-source implementation of RV-aware modules of ANN layers. Specifically, RV-Linear, RV-Convolution, RV-Transposed Convolution, RV-Batch Normalization, and RV-ReLU activation function.
- The modification and open-source implementation of RV-VAE family of models.
- The design and implementation of SRV-VAE family of models.

Appendix B
Additional Utilization of RV modules

Moreover, outside the context of this thesis, an additional system was developed for data augmentation utilization. This system employed RV modules presented in Chapter 3 to treat augmented parts of an image as continuous distributions instead of data samples.

Specifically, we created a small toy network of 3 hidden layers (consisting of Convolution followed by a ReLU) and a Fully Connected layer as an output for the task of classification of hand-written digits. MNIST [83] dataset was used with an augmented background with samples drawn from a Uniform distribution $\mathcal{U}(0, 1)$. During training, we trained two variants of the toy model, one regular (T) and one with RV modules (RV-T). The T model was trained with samples of the Uniform distribution as background, while RV-T instead of samples/pixels was trained with RVs as input.

In Tab. B.1 we report the accuracy of each model after being trained for 100 epochs in a specific percentage of the MNIST training set. As we can observe, for smaller training set percentages the RV variant is better, and even by a large margin in the last case, while not being notably worse in the cases of larger training set percentages. We should also state here, that the T models required 20-30 epochs to converge to the optimal accuracy, while our RV-T variants needed only 3-10 epochs to converge.

These results justify the beneficial use of RV modules in cases where continuous representations of probabilistic quantities can be used instead of sample estimations, proving once more the strength of RV modules utilization in ANNs.

Table B.1: Classification accuracy of the two toy networks on the modified MNIST test sets when trained on different percentages of the training set.

Training set %	T model	RV-T model
100%	**99.12%**	99.06%
75%	**98.94%**	98.86%
50%	98.31%	**98.71%**
25%	81.93%	**95.81%**

Appendix C
Acronyms

AR Augmented Reality 4

ANN Artificial Neural Network 1

BNN Bayesian Neural Network 35

CDF Cumulative Distribution Function 39

CNN Convolutional Neural Network 1

CRF Conditional Random Field 14

DNN Deep Neural Network 1

ELBO Evidence Lower Bound 45

FID Fréchet Inception Distance 49

FLOPS Floating Point Operations per Second 47

GAN Generative Adversarial Network 31

HCI Human-Computer interaction 10

IoU Intersection over Union 24

LLM Large Language Model 1

LVM Large Vision Model 1

MPJPE Mean per Joint Positional Error 73

MRF Markov Random Field 11

MSE Mean Squared Error 20

PDF Probability Density Function 31

ReLU Rectified Linear Unit 19

RGB Red Green Blue channel 1

RGBD Red Green Blue Depth channel 2

RHD Rendered Handpose Dataset 22

RV Random Variable 32

RV-VAE Random Variable Variational Autoencoder 32

S-VAE Supervised Variational Autoencoder 67