UNIVERSITY OF GENOVA

PHD PROGRAM IN BIOENGINEERING AND ROBOTICS

Advancing Core Components of Robotic Manipulation

New Methods for 3D Perception, Semantic Understanding, and Policy Learning

by

Andrea Rosasco

Thesis submitted for the degree of *Doctor of Philosophy* (38° cycle)

March 2026

Lorenzo Natale — Supervisor

Giulia Pasquale — Supervisor

Paolo Massobrio — Head of the PhD program

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

I dedicate this thesis to those who stood by me.
Resilience grows in good company, and I was lucky to have yours.
Every late night was worth it because of your support.
My gratitude is deeper than these words can show.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Andrea Rosasco
March 2026

Acknowledgements

My deepest thanks go to Dr. Lorenzo Natale for the opportunity to pursue my Ph.D. at Istituto Italiano di Tecnologia within the Humanoid Sensing and Perception group. Over these years, I have deeply valued his mentorship and the trust he placed in me to explore and develop my research ideas.

I am grateful to Dr. Giulia Pasquale and Dr. Federico Ceola for mentoring me and teaching me the fine art of publishing scientific research. Your guidance has shaped me as a researcher.

A special thanks to Stefano, Vignesh, Simone, Michele, and Jon – a.k.a. the ergoCub team – for all the work we did together, from developing our demos to presenting them at countless events. Working with you has been an honor.

Finally, I would like to thank all the colleagues in HSP for all the conversations, jokes, and experiences we shared. Without you, these years would certainly have been more boring. And to all who spent entire nights at the lab with me for some conference deadline – may we never do it again.

Abstract

Robotic manipulation in unstructured, human-centric environments presents a significant challenge for autonomous systems. A robot must be able to semantically perceive the environment, understand its 3D structure, and generate effective motions to complete a task. Each of these stages corresponds to distinct research areas, each with its own set of challenges. The goal of this thesis is to study existing approaches, understand their weaknesses, and propose novel state-of-the-art algorithms. An important part of the evaluation of the developed algorithms is their deployment on robotic platforms. The thesis is structured around three key components of robotic manipulation: 3D perception, semantic understanding, and motion generation. For each component, I identified the limitations of existing methods with respect to the target evaluation setting. I then developed novel data-driven algorithms and benchmarks to overcome these limitations and advance the state of the art in each respective area. The research conducted has led to three main contributions. First, to address the challenge of 3D perception from partial sensor data, I developed a confidence-guided shape completion algorithm. This method leverages a transformer-based HyperNetwork architecture to reconstruct objects at arbitrary resolutions while producing a confidence estimate over its output. Second, to improve semantic perception, particularly for novel or user-specified objects, I collaborated on the creation of ConCon-Chi, a benchmark for personalized vision-language tasks. This benchmark challenges existing models to learn new concepts from a few images and to compose them with known contexts, providing a robust framework for assessing concept-context compositionality. Third, to enhance robot motion generation, I proposed KDPE, a kernel density estimation-based strategy to improve the safety and reliability of policies learned via behavior cloning. KDPE filters out non-representative action trajectories generated by diffusion models at inference time, improving performance and stability both in simulation and on real-world robotic platforms. The algorithms proposed throughout this thesis were integrated into modular and end-to-end manipulation systems and extensively tested through demonstrations and experimental evaluations. This practical validation allowed me to gather insights into the trade-offs of each approach under different aspects, such as robustness, interpretability, computational cost, and flexibility. Ultimately, this thesis

provides a validated set of algorithms and benchmarks that can be used to build more capable and reliable robotic systems for complex manipulation tasks.

Table of contents

List of publications

Included Publications

- Rosasco, Andrea and Berti, Stefano and Bottarel, Fabrizio and Colledanchise, Michele and Natale, Lorenzo. "Towards Confidence-guided Shape Completion for Robotic Applications". *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 580-586.

- Rosasco, Andrea and Berti, Stefano and Pasquale, Giulia and Malafronte, Damiano and Sato, Shogo and Segawa, Hiroyuki and Inada, Tetsugo and Natale, Lorenzo. "ConCon-Chi: Concept-Context Chimera Benchmark for Personalized Vision-Language Tasks". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 22239-22248.

- Rosasco, Andrea and Ceola, Federico and Pasquale, Giulia and Natale, Lorenzo. "KDPE: A Kernel Density Estimation Strategy for Diffusion Policy Trajectory Selection". *Proceedings of The 9th Conference on Robot Learning (CoRL)*, 2025, pp. 1210-1224.

Excluded Publications

- Berti, Stefano and Rosasco, Andrea and Colledanchise, Michele and Natale, Lorenzo. "One-Shot Open-Set Skeleton-Based Action Recognition". *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022, pp. 765-772.

List of figures

List of tables

List of acronyms

Acronyms / Abbreviations

CNN	Convolutional Neural Network
DB	DreamBooth
DDIM	Denoising Diffusion Implicit Models
DDPM	Denoising Diffusion Probabilistic Model
DP	Diffusion Policy
GPD	Grasp Pose Detection
GT	Ground Truth
KDE	Kernel Density Estimation
LDM	Latent Diffusion Model
mAP	mean Average Precision
mRR	mean Reciprocal Rank
NLP	Natural Language Processing
PDF	Probability Density Function
PerVL	Personalized Vision & Language
RL	Reinforcement Learning
SDM	Stable Diffusion Model
TIG	Text-to-Image Generation

TIR	Text-to-Image Retrieval
TI	Textual Inversion
ViT	Vision Transformer
VL	Vision-Language
YCB	Yale-CMU-Berkeley (Object Set)
ZS-CIR	Zero-Shot Composed Image Retrieval

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

Historically, robots have existed in the form of very specialized machines confined to controlled industrial scenarios. The flexible and generalist robot archetype often depicted in science fiction only existed in rudimentary form in research labs, held back from widespread application by hardware limitations and compute costs. As robotic technology evolves, the prospect of multi-task robots operating in unstructured environments is beginning to gain traction, shifting the focus from hardware to algorithms capable of performing well across a variety of settings. In industrial applications, robots typically perform repetitive tasks under controlled environmental conditions and with full information. In contrast, unstructured environments such as households, hospitals, and warehouses are characterized by partial observability, unpredictability, and potential human interaction. To operate in these settings, robots need algorithms that are either robust to distribution shift or easily adaptable to new environments. This is particularly true in the context of robotic manipulation, where the reliance on images to complete complex tasks can easily expose the model to distribution shift between training and deployment observations. This thesis focuses on advancing robotic manipulation in unstructured environments by addressing limitations in three connected research fields that enable it: 3D perception, semantic understanding, and motion generation. The importance of these fields for manipulation in unstructured environments arises from the necessity of a robot to map image observations, and possibly a task specification, to a series of actions that control the robot to complete the task. For example, starting from an image, a manipulation algorithm has to identify the object of interest for the task, estimate its geometry and relative position in the world and generate a series of actions to interact with it. Whether these steps are implemented as separate modules or implicitly learned by an

end-to-end model, they remain key aspects of robotic manipulation, each with its own set of challenges.

To interact with objects in the real world, a robot needs to know their geometry in 3D space. Partial 3D information can be obtained through the use of depth sensors such as LiDARs and stereo cameras. However, depth information collected from a single viewpoint is incomplete due to object self-occlusions. A popular approach to recover the missing information is via object pose estimation. First, the 6D pose of the object is estimated by a data-driven method that usually takes as input an RGB-D image, and then a 3D model of the object is aligned to the estimated pose to reconstruct the full 3D geometry (Tremblay et al., 2018; Xiang et al., 2018; Zakharov et al., 2019). While this approach can potentially reconstruct the object with a high degree of accuracy, its reliance on a 3D model of the object limits its deployment to novel domains. Recent works (Barad et al., 2024; Wen et al., 2023) alleviate this limitation by jointly tracking object pose and reconstructing an object model *online*, but they typically rely on multiple views (i.e., camera/object motion) to recreate a model of the object. A more flexible approach is shape completion. Shape completion algorithms take as input the partial observation of the object and predict an estimate of its full geometry without requiring a pre-computed CAD model. While shape completion is widely studied in 3D vision, its integration into robotic grasping has been comparatively less explored, and existing methods often struggle to generalize to real-world observations. Therefore, there is a strong motivation to develop shape completion algorithms for robotics that are accurate and general.

Completing complex tasks requires being able to recognize objects based on their appearance and description. Even though deep learning models can be trained to recognize an impressive number of objects (Minderer et al., 2023; Radford et al., 2021), they are limited to the ones seen during training and cannot easily adapt to new ones. As robots are deployed in new and dynamic environments, it is crucial for them to be powered by algorithms that are easily personalizable. The personalization of vision-language models has been recently explored in the context of image retrieval (Baldrati et al., 2023; Cohen et al., 2022; Saito et al., 2023) and image generation (Gal et al., 2023a; Ruiz et al., 2023), and refers to adapting a pre-trained model to learn novel user-specific concepts from a few examples without degrading its previously acquired knowledge. The model should then learn to recognize, or to generate, images containing the novel concept in known varied contexts. While mostly applied to retrieval and generation, personalization methods can also be relevant to robotics: a robot may be asked to interact with a previously unseen object, such as a newly introduced tool or a user-specific item, that was not part of the model

pre-training data. In these cases, personalization techniques are necessary for the model to recognize and interact with the object. However, existing benchmarks for personalized image retrieval present several shortcomings that limit their reliability in assessing vision-language models' performance. More challenging benchmarks to thoroughly evaluate methods for the personalization of vision-language models are necessary to push the field forward. Moreover, personalization benchmarks for image retrieval could be used to assess the personalization of robotic systems by probing the intermediate layers of a deep robotic policy (Wang et al., 2024) to determine whether new concepts are learned and recognized correctly.

Finally, to get a robot to perform a task, an algorithm needs to generate and stream control commands to its motors. For simple tasks such as pick-and-place, this can be done by localizing the object and sending target poses to a Cartesian controller. However, for tasks requiring complex logic and movements, such as folding a T-shirt, a more complex approach is necessary. In recent years, behavior cloning has been successfully applied to robotic manipulation (Florence et al., 2022; Mandlekar et al., 2021; Shafiullah et al., 2022). Behavior cloning can be used to train models that are capable of learning complex behaviors from a limited number of demonstrations. These models, also known as visuomotor policies, work by taking as input images and proprioception from the robot sensors and outputting Cartesian or joint target positions. These algorithms are run in closed-loop until task completion. While this allows the robots to exhibit reactive behaviors, it also means that errors in the policy output can compound over time, possibly leading to task failure. This risk can be more pronounced for stochastic generative policies such as diffusion policies (Chi et al., 2023): since, at inference time, actions are generated by sampling from the conditional distribution learned by the model, outlier samples can steer the model away from the known distribution, exacerbating compounding error.

By addressing these specific challenges in research fields that are key to robotic manipulation, this thesis aims to improve the algorithmic building blocks necessary for the next generation of autonomous manipulation systems.

1.2 Research Objectives

The objective of this thesis is to tackle the challenges outlined in Section 1.1 through the development of novel algorithms and benchmarks. Where appropriate, the proposed methods have been deployed on real-world robotic platforms, namely the ergoCub humanoid robot and the Franka Emika Panda robotic arm (Haddadin et al., 2022).

Regarding 3D perception, I worked on shape completion for robotic manipulation as part of the ergoCub project. My goal in this area was to develop a shape completion algorithm that generalizes to novel objects, improving on both standard completion metrics and grasp success when integrated into a grasping pipeline.

Following my work on shape completion, I reviewed methods for the personalization of vision-language models as a way of identifying objects for the robot to interact with. While multiple methods exist in the context of personalized image retrieval, the benchmarks used to evaluate them presented several shortcomings that limited their reliability. To tackle this, I collaborated on the creation of ConCon-Chi (Rosasco et al., 2024) with the objective of providing a benchmark to better evaluate the performance of personalization methods for vision-language models.

Finally, I focused on visuomotor policies trained to generate trajectories of actions that can be executed on a robot to complete manipulation tasks. My goal was to increase the reliability of generative policies through the use of test-time computing, improving the performance of pre-trained policies without the need for retraining them.

1.3 Thesis Outline

This thesis is structured as follows:

- Part II presents my scientific contributions, which consist of three published papers, whose most relevant details are summarized in Chapter 2, Chapter 3, and Chapter 4, and the implementation of a demo on the ergoCub humanoid robot, described in Chapter 5. The paper summaries are intended to provide a minimal background while highlighting the key contributions of the papers.

- The three scientific papers that constitute the core contributions of this thesis are included in Part III as revised versions of the published papers.

- Finally, in Part IV, I provide an overview of the work presented in this thesis, discuss its limitations, and outline potential directions for future research. In particular, when discussing future research directions, I highlight possible synergies between the three research areas explored in this thesis.

Part II

Contributions

Chapter 2

Confidence-guided Shape Completion

My contribution in the field of 3D perception for robotic manipulation focused on shape completion of point clouds and resulted in the publication *Towards Confidence-guided Shape Completion for Robotic Applications* (Rosasco et al., 2022). The full publication is provided in Chapter 6.

Grasp detection in unstructured environments can benefit from working on the object's complete 3D information. Normally, the 3D information obtained from a single point of view is incomplete due to self-occlusion but it can be reconstructed using shape completion. While shape completion is widely studied in 3D vision, its integration into robotic grasping pipelines has been comparatively less explored, and existing methods often struggle to generalize to real-world observations. To address these limitations, I propose a novel confidence-guided shape completion method consisting of a transformer-based HyperNetwork that generates an implicit function representing the reconstructed object. Then, through a novel gradient-based sampling strategy, a point cloud of high-confidence points can be extracted from the implicit function and used for grasp detection.

The method was validated in terms of shape completion accuracy and grasping success rate when used as part of a robotic grasping pipeline.

2.1 Background

Shape completion is the task of reconstructing the full 3D geometry of an object given a partial observation. This task finds applications in medicine (Guo et al., 2018), computer-aided design (Li et al., 2022b) and robotics (Varley et al., 2017). While classical methods for shape completion have tried to recreate the missing geometry by fitting geometric primitives (Duncan et al., 2013; Schnabel et al., 2009) or by retrieving similar shapes from a database

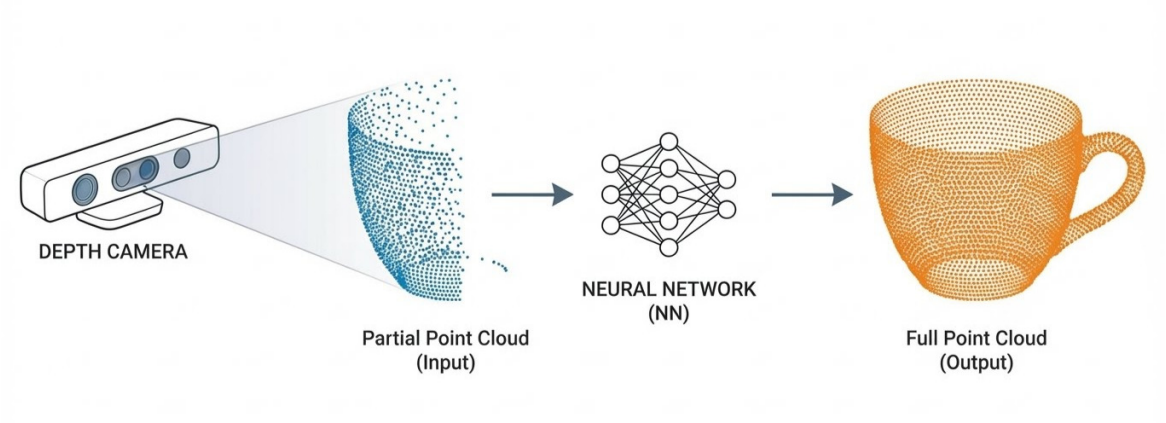


Figure 2.1 Object self-occlusion results in partial observations (left) that can be completed using shape completion methods (right).

(Pauly et al., 2005; Sung et al., 2015), more recently, data-driven methods based on deep learning have become the de facto standard (Choy et al., 2016; Dai et al., 2017). These methods are trained on large datasets of 3D models such as ShapeNet (Chang et al., 2015) and ModelNet (Wu et al., 2015) and show impressive results in the reconstruction of known and unknown objects. Deep learning methods for shape completion can be categorized based on the 3D representations they use, such as voxels, point clouds and implicit representations. In this chapter I outline some general concepts and architectures related to shape completion. I also present a grasp detection algorithm that works on point clouds and can benefit from shape completion.

2.1.1 3D Representations

- **Voxels**: 3D models are represented as a 3D occupancy matrix where each value, called a voxel (volumetric pixel), can be either empty or occupied. An advantage of this representation is that Convolutional Neural Networks (CNNs) can be easily extended to work on voxel grids (Choy et al., 2016; Wu et al., 2018). However, increasing the resolution of the object representation causes the voxel grid size to increase cubically, leading to high memory and computational requirements.

- **Point Clouds**: 3D models are represented as a set of points in the 3D space. Point clouds are a flexible representation that can easily represent complex geometries at high resolution. While CNNs cannot be directly applied to point clouds, different ways of processing them have been proposed (Qi et al., 2017; Wang et al., 2019). One drawback of point clouds is that, contrary to voxel grids, when used as the output

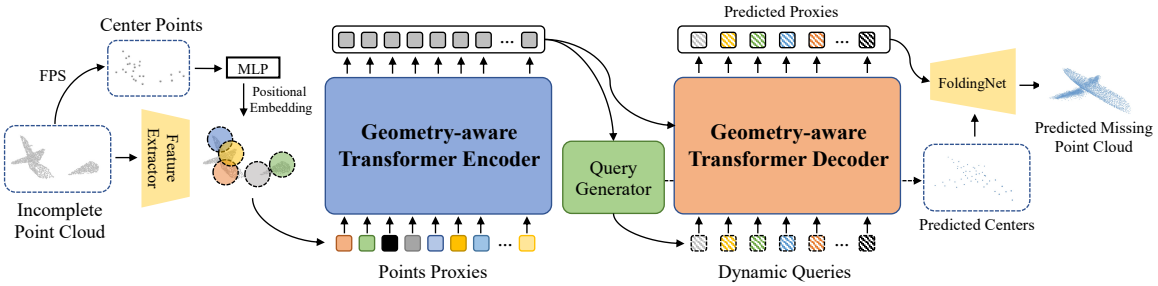


Figure 2.2 PoinTr architecture overview. Source: Yu et al. (2021).

representation of a neural network, they do not carry information about the confidence of the prediction.

- **Implicit Representations**: 3D models are represented as continuous functions that define the surface of the object. Implicit representations can represent complex geometries at arbitrary resolution and can be efficiently stored (Park et al., 2019). However, an implicit function is often not suitable for downstream tasks, such as grasp detection, and needs to be converted to explicit representations such as meshes or point clouds.

2.1.2 Architectures for Shape Completion

PoinTr

After its proposal in Vaswani et al. (2017), the transformer architecture has been successfully applied in natural language processing (Devlin et al., 2019) and computer vision (Parmar et al., 2018). More recent work managed to adapt transformers to process point clouds allowing for their application to 3D shape completion. A seminal work in this direction is PoinTr (Yu et al., 2021). PoinTr treats shape completion as a sequence-to-sequence problem where the input sequence is the partial point cloud and the output sequence is the completed point cloud. To reduce the input point cloud to a sequence of tokens, a specialized layer called Dynamic Graph CNN (DGCNN) (Wang et al., 2019) is applied to the input point cloud to extract local features. Each embedding extracted by DGCNN represents a local patch of the input point cloud and is associated with a 3D position which is used to compute the positional encoding. The sum of the positional encodings with the patch embeddings is processed by an encoder-decoder transformer architecture to produce a sequence of point proxies representing the completed shape. Finally, in a coarse-to-fine fashion, the point proxies are decoded into the complete point cloud by applying FoldingNet (Yang et al., 2018), a decoder which deforms a 2D grid of points into the final 3D surface. The attention

mechanism used in PoinTr is biased on the local geometry of the point cloud by attending to the k-nearest neighbors of each point proxy. The architecture of PoinTr is shown in Figure 2.2.

HyperNetwork

HyperNetworks (Ha et al., 2017) are a class of neural networks that generate the weights for another neural network, called the target network. This type of architecture presents several advantages such as the ability to adapt the target network to different inputs and the reduction of the number of parameters. Due to these advantages, HyperNetworks have been applied to various tasks such as few-shot learning (von Oswald et al., 2020), image processing (Chauhan et al., 2024) and 3D shape generation (Littwin and Wolf, 2019). Recent research attributes the compositional generalization performance of transformers to their implicit use of HyperNetworks (Schug et al., 2025).

Formally, let $f_\theta : \mathscr{X} \to \mathscr{Y}$ denote the target network with parameters $\theta \in \Theta$, mapping an input x (e.g., spatial coordinates) to an output y. Let $h_\phi : \mathscr{Z} \to \Theta$ be the HyperNetwork mapping a conditioning signal $z \in \mathscr{Z}$ (e.g., input or task embedding) to the parameters of f_θ. The resulting prediction is given by

$$F_\phi(x,z) = f_{\theta_z}(x) = f_{h_\phi(z)}(x). \tag{2.1}$$

Given a data distribution $p_{\text{data}}(x,y,z)$ and loss $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}_+$, learning reduces to

$$\min_\phi \mathscr{L}(\phi) = \mathbb{E}_{(x,y,z)\sim p_{\text{data}}} \left[\ell(f_{h_\phi(z)}(x), y) \right], \tag{2.2}$$

which is optimized via stochastic gradient descent methods using the empirical estimate of $\mathscr{L}(\phi)$.

2.1.3 Grasp Pose Detection (GPD)

In my publication on shape completion, to perform grasping experiments, I paired my algorithm with GPD (ten Pas et al., 2017a). GPD is a method for grasp detection in point clouds. The algorithm takes as input the point cloud of an object and outputs a 6D grasp pose. First, a set of grasp candidates is sampled on the point cloud surface and filtered according to geometric criteria. Then, each candidate is encoded as an image and is processed by a CNN trained to predict whether the grasp pose is a force closure grasp or not. Finally, the remaining grasps are ranked according to geometric heuristics and the highest ranked one is

executed. The authors show that the performance of GPD is affected by the density of the point cloud, achieving better results when an object is observed from multiple viewpoints and presents fewer occlusions.

2.2 Method

2.2.1 Architecture

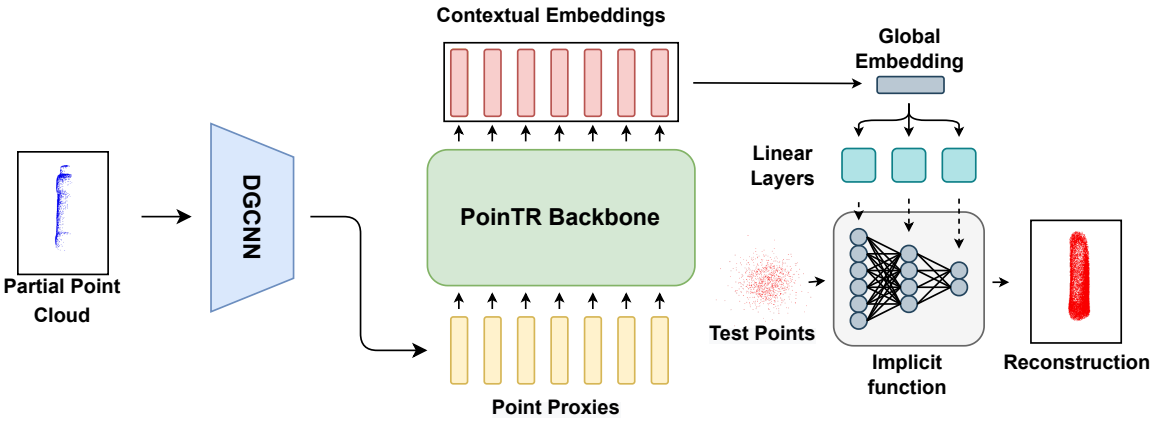


Figure 2.3 Overview of the proposed shape completion architecture. The backbone computes the contextual embeddings of the input point proxies. Then, the embeddings are passed through a max-pooling layer to obtain the global embedding. Finally, the global embedding is fed to different linear layers to generate weights, biases and scales of the implicit function.

The proposed method leverages the encoder of the transformer-based architecture PoinTr presented in Yu et al. (2021) (see Section 2.1.2) to process the partial input point cloud and extract a global shape embedding. Unlike PoinTr, which uses a decoder to attend to the encoder output embeddings and generate a fixed-size point cloud, my method computes a global embedding e by applying a max-pooling layer on the encoder's output features. Then, following the HyperNetwork paradigm (Ha et al., 2017) explained in Section 2.1.2, the embedding e is passed through a set of linear layers to generate the weights of an MLP which constitutes the implicit function representation of the reconstructed object. Specifically, the weights of the MLP are computed by processing e through a set of linear layers. Following the work of Littwin and Wolf (2019), for each layer l of the MLP I generate three sets of weights: W_l, b_l and s_l, which represent respectively the weights, biases and scales of layer l.

Given an input point $x_1 = p$, the output of the implicit function with L layers is $y = x_L$, where the output of each layer is defined as:

$$x_{l+1} = \sigma\left(((\phi_l^W x_l) \cdot \phi_l^s) + \phi_l^b\right), \tag{2.3}$$

where σ is the Leaky ReLU function on all the hidden layers and the sigmoid function on the output layer.

2.2.2 Training

To train the model, the backbone takes as input the partial point cloud X and generates the weights ϕ of the implicit function g_ϕ. Then g_ϕ is queried on a batch of points P_Y sampled as follows: 50% of the points are sampled from the surface of the ground truth object Y, 40% are sampled around the object surface, and 10% are uniformly sampled within the output space. For each query point $p \in P_Y$, the ground truth occupancy value $o(p)$ is defined based on its distance from Y. Specifically, $o(p) = 1$ if the distance from p to the nearest point in Y is below a threshold τ, and $o(p) = 0$ otherwise. The training objective is to minimize the binary loss between the implicit function and the ground truth values

2.2.3 Gradient-based Sampling

I propose an algorithm to sample a point cloud from an implicit occupancy function exploiting the implicit function's output confidence. First, an initial set of points P_{init} is uniformly sampled within the output space. Then, the points are iteratively updated by gradient descent, optimizing them to maximize the output of the implicit function. At every step, all points with confidence above a threshold are added to the output point cloud. This process is repeated for a fixed number of iterations and is able to quickly generate a full point cloud of high confidence points.

Compared to the naive approach of passing a grid of points through the implicit function and keeping the ones above a certain confidence threshold, my approach manages to create a dense point cloud of high confidence points more efficiently.

2.3 Experimental Validation

The proposed method was validated through two metrics: the shape completion accuracy measured via Jaccard similarity and the grasping success rate obtained when using the algorithm's output as the input to a grasp detector.

2.3.1 Shape Completion Performance

I compared my method against two state-of-the-art shape completion methods for robotics: Varley et al. (2017) (V) and Lundell et al. (2019) (USN). The performance was measured by computing the Jaccard Similarity between the ground truths and the reconstructions. The methods were trained on a dataset of self-occluded and complete object point clouds generated from 18 models from the YCB object set (Calli et al., 2015) and 590 models from the Columbia Grasp Database (Goldfeder et al., 2009). Following the same protocol as V and USN, I reported the performance on three different splits: *Training Views*, *Holdout Views* and *Holdout Models*, where *Holdout Views* contains occlusions of objects that can be found in the training set, while *Holdout Models* contains occlusions of objects that were excluded from the training set. As shown in Table 6.1, my method outperforms both V and USN in all three splits.

2.3.2 Robotic Grasping with GPD

To demonstrate the practical applicability of my method, I created a modular grasping pipeline that could integrate different shape completion methods. The pipeline first segments the input point cloud using RANSAC to obtain the partial object point cloud. Then, the partial point cloud is reconstructed using the shape completion algorithm and its output is passed to GPD (Section 2.1.3) to generate a grasp proposal. Finally, GraspIt (Miller and Allen, 2004) is used to simulate and execute the grasp on a Franka Emika Panda arm equipped with a Robotiq 2F-85 gripper. For the experiment I used a set of 10 unseen objects shown in Figure 6.3. I attempted a total of 10 grasps per object on five different orientations (0°, 72°, 144°, 216°, and 288°) for a total of 100 grasps per method.

While my method struggles with some of the more complex objects, on average it performs better than USN achieving a 10% increase in success rate. The results on each test object are reported in Figure 6.6.

Chapter 3

The ConCon-Chi Benchmark

In this chapter, I present my contributions to the field of vision-language personalization which were published by my coauthors and me in the paper *ConCon-Chi: A Benchmark for Evaluating Concept-Context Compositionality in Personalized Vision-Language Models* (Rosasco et al., 2024). The full publication is provided in Chapter 7.

Recent benchmarks for personalized image retrieval and image generation suffer from two key limitations: they often only contain common concepts, reducing the task to synonym matching, and lack a rich compositional structure, causing context bias. I propose ConCon-Chi, a benchmark designed to address these limitations by (i) introducing chimeric concepts to evaluate the learning of novel meanings and (ii) containing a substantial overlap of concepts and contexts to reduce concept and context bias. This summary primarily focuses on the aspects related to image retrieval as it is more relevant to robotics applications. In fact, both in the context of image retrieval and of language-conditioned robotic policies, a model needs to generate a meaningful embedding that encodes the necessary information to recognize an object. For this reason, the ConCon-Chi benchmark could find applications in the evaluation of language-conditioned manipulation policies. The experiments and analyses related to image generation can be found in Section 7.5.3 and Section 7.5.4.

3.1 Background

Vision-language (VL) models are multimodal architectures capable of processing both images and text. They are usually trained on internet-scale datasets allowing strong generalization performance at the cost of a long and expensive training. Because of this, adding new knowledge after training is particularly challenging: retraining on the original data is often

unfeasible, and fine-tuning only on the new one would result in the catastrophic forgetting of the previously learned concepts.

Vision-Language Personalization aims to teach vision-language foundation models novel concepts using a limited number of image examples. There are two important requirements for personalization methods:

- **Compositionality**: the method should be able to combine the learned novel concepts with a *context*: a text description that specifies how the novel concept should appear in the retrieved or generated images. For example, the context could specify the scene where the object should appear, its color, its size, etc.
- **Robustness to catastrophic forgetting**: the method should be able to learn novel concepts without degrading the performance of the pre-trained model on previously learned concepts.

In the remainder of this chapter, I introduce two representative application settings: personalized text-to-image retrieval and personalized text-to-image generation.

3.1.1 Text-to-Image Retrieval

The task of text-to-image retrieval (TIR) has been widely studied in computer vision and natural language processing for its broad applications in various domains such as healthcare (Ou et al., 2025), e-commerce (Wang et al., 2023), and multimedia search (Long et al., 2024). The goal of image retrieval consists of finding the correct images from an image pool given a text query. More specifically, given a dataset of images $\{I_1, I_2, \ldots, I_N\}$ and a text query T, the goal is to produce a ranking for the images in the image pool such that the most relevant images to the text query are ranked higher. In recent years, large-scale pre-trained Vision-Language models, such as CLIP (Radford et al., 2021), have achieved impressive results on this task. These models are trained on large datasets of image-text pairs and learn to map images and texts into a shared embedding space where semantically similar images and texts are close to each other. Image retrieval is then performed by computing the similarity between the text query and the images in the embedding space and ranking the images based on their scores (Figure 3.1).

More formally, let E_I and E_T be the image and text encoders respectively. Given an image I and a text T, the model computes the image embedding $v_I = E_I(I)$ and the text embedding $v_T = E_T(T)$. The similarity score between the image and the text is then computed as the

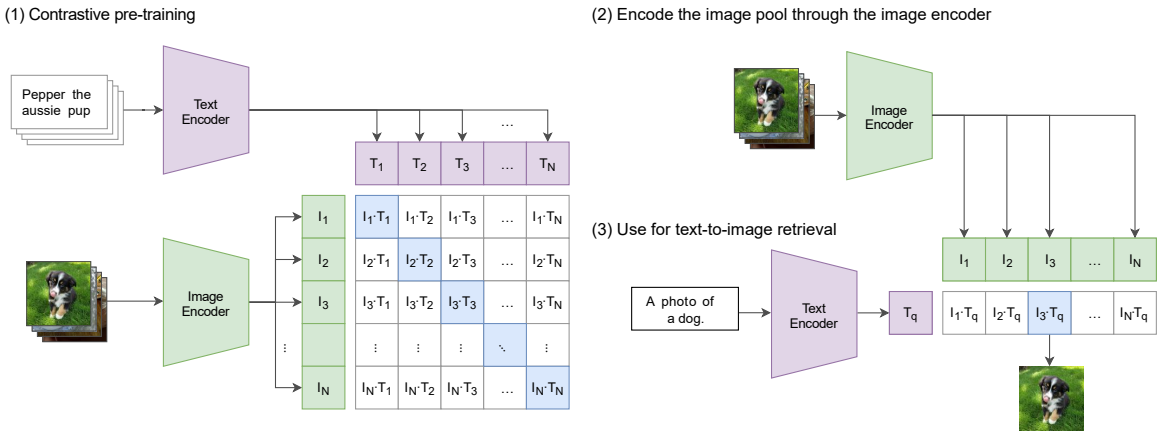


Figure 3.1 CLIP is trained by minimizing the cosine distance between the image and text embeddings of matching image-text pairs while maximizing the distance between non-matching pairs. The trained CLIP encoders can be used to perform image retrieval by computing the embedding of the image pool (2) and comparing it with the embedding of the query (3). Adapted from Radford et al. (2021).

cosine similarity between the two embeddings:

$$s(I,T) = \frac{v_I \cdot v_T}{\|v_I\|\|v_T\|}. \tag{3.1}$$

During training, the model is optimized to maximize the similarity score for matching image-text pairs while minimizing it for non-matching pairs using a contrastive loss function.

While pre-trained vision-language models achieve impressive results on image retrieval tasks, their performance inevitably degrades when the text queries describe novel concepts that were not well represented in the pre-training data.

3.1.2 Text-to-Image Generation

Text-to-Image Generation (TIG) is the task of generating an image that corresponds to a given text description. Recent advancements in generative models, particularly Diffusion Models (Ho et al., 2020), have led to algorithms capable of generating high-quality, diverse images from text prompts (Rombach et al., 2022; Saharia et al., 2022). These models are typically trained on massive datasets of image-text pairs and learn to reverse a diffusion process, gradually denoising a random noise vector to produce a coherent image.

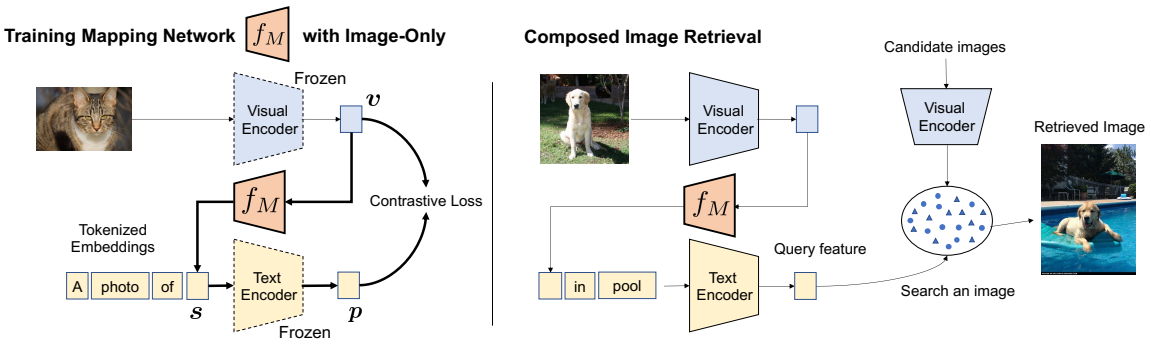


Figure 3.2 The Pic2Word personalization algorithm works by freezing the CLIP encoders and learning a mapping function from the visual encoder output space to the text encoder input space by optimizing CLIP's contrastive loss. After the mapping function has been trained, it can be used to map images to text tokens that can be used for retrieval. Source Saito et al. (2023).

3.1.3 Personalized Text-to-Image Retrieval

Personalized image retrieval consists of learning a user-specific *concept* using a few images and then performing retrieval of such *concept* in a known *context*. Since personalized image retrieval methods are based on pre-trained vision-language models, the benchmarks used to evaluate them do not include a large amount of training data. Instead, they are usually structured as follows: an image pool $\mathscr{P}$ containing images of the novel concepts in different contexts, a set of text queries $\mathscr{Q}$ relating concepts to contexts and a small set of images $S_k = \{x_1, \ldots, x_k\}$ for each novel concept. A variant of personalized TIR where only one image per concept is provided is called Zero-Shot Composed Image Retrieval (ZS-CIR).

A common approach to personalized image retrieval consists in learning a mapping ϕ between images and the text encoder input space. In this way, images of the novel concepts in the support set S_k can be mapped to one or more text tokens $v^* = \phi(S_k)$ that can be directly combined with the tokens representing the context and processed by the text encoder to produce the final text embedding used for retrieval. An example of this approach comes from Pic2Word (Saito et al., 2023) described in Figure 3.2.

3.1.4 Personalized Text-to-Image Generation

Personalized Text-to-Image Generation (TIG) was proposed in Gal et al. (2023a) and is the generative counterpart of personalized TIR. The task consists of learning a user-specific *concept* using a few images and then performing generation of such *concept* in a known *context*. Textual Inversion (Gal et al., 2023a) expands the vocabulary of a frozen text-to-image

model (Latent Diffusion (Rombach et al., 2022)) with user-specific concept embeddings learned from a few images. DreamBooth (Ruiz et al., 2023) instead selects a rarely-used token and binds it to the concept by fine-tuning the entire text-to-image model on the concept images. Subsequent methods propose improvements over these two (Gal et al., 2023b; Kumari et al., 2023; Tewel et al., 2023).

3.2 Design Principles

The design principles that guided the creation of ConCon-Chi reflect the limitations found in existing benchmarks. In particular, most personalization benchmarks suffer from concept and context bias, and do not evaluate the learning of truly novel concepts. ConCon-Chi addresses these shortcomings through two design principles: its grid structure and the use of chimeric concepts.

3.2.1 Grid Structure

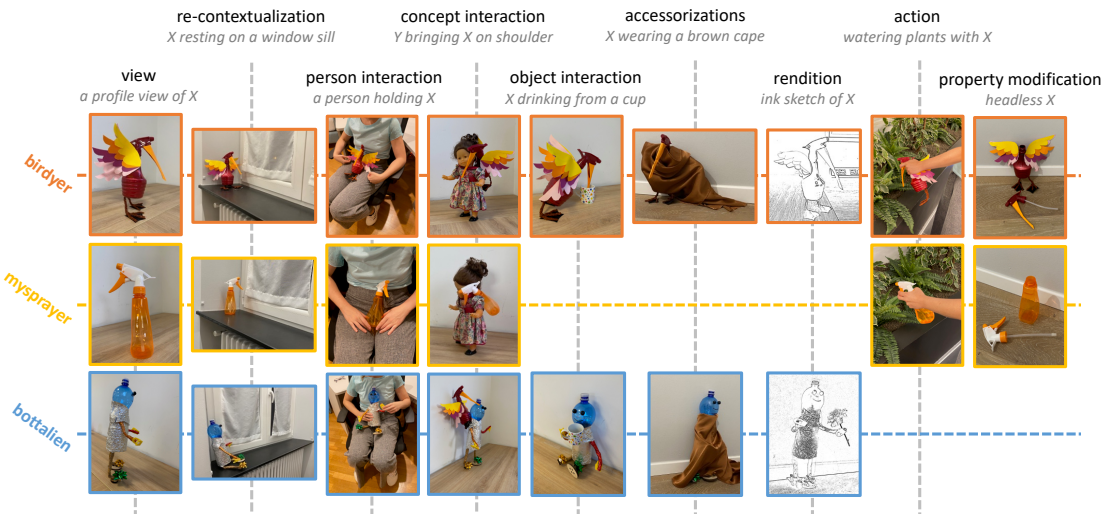


Figure 3.3 **Concept-context structure of the ConCon-Chi dataset.** Excerpt from the dataset structure (see Figure 7.4). Each row represents images of a concept (concept name in Color) and each column images of a context (context kind in Black and context in Grey). The caption for an image is formed by the composition of the context with the concept (replacing X/Y with the concept textual identifier).

Many benchmarks for personalized image retrieval suffer from concept and context bias. Context bias occurs when in a benchmark the majority of contexts are combined with only one or a small number of concepts. This allows the personalization method to score high

performance without learning the new concept itself, by simply retrieving the image based on the context. Similarly, concept bias occurs when concepts are combined with a low number of contexts and has symmetrical implications. I computed the average number of concepts per context and contexts per concept for several benchmarks showing how most benchmarks have a low value for both metrics (see Table 7.1). The two exceptions are *CC101* and *DreamBooth* (Kumari et al., 2023; Ruiz et al., 2023). However, being personalized image generation benchmarks they do not have ground truth images but just a collection of prompts relating concepts to contexts. To minimize concept and context bias, ConCon-Chi was created by finding contexts that could be combined with a large number of concepts and vice versa. This resulted in a *concept-context grid* (Figure 3.3) that was used to guide the data acquisition process. For every feasible concept-context pair an average of 6 images depicting the concept in the specified context was manually collected.

3.2.2 Chimeric Concepts

A key aspect of personalization is the ability to learn novel concepts from a few examples. However, given that personalization methods are based on foundation models pre-trained on internet-scale datasets, it is challenging to find concepts that are truly novel to the model. Existing benchmarks either use common concepts, placing their focus on the composition of concept and context rather than the learning of new concepts, or use fine-grained concepts (e.g. specific fashion items) that might still be present in the pre-training dataset.

Existing objects, whether they are common or rare, have been probably observed during pretraining. For this reason, inspired by Lazaridou et al. (2017), ConCon-Chi contains a set of 6 *chimeric* concepts obtained by mixing features of existing concepts. All the concepts can be seen in Figure 7.2. Since the training set of foundation models is often not disclosed it is not possible to tell to which extent a concept is novel. However, using novel handcrafted concepts ConCon-Chi minimizes the chance that the model has already observed them during pre-training.

3.3 Experimental Validation

3.3.1 Concept-Context Bias

To study how concept and context bias affect different benchmarks, I reproduced an experiment proposed in Saito et al. (2023). I implemented a baseline retrieval method that computes the query embedding as a weighted sum of the concept and context embeddings extracted

from CLIP image and text encoders respectively. Then I plotted the mAP of the baseline method as a function of the weight, α, for all the personalized TIR and ZS-CIR benchmarks reported in Table 7.1. The resulting plot (Figure 7.5) shows how, for most benchmarks, the decrease in performance caused by only using the context ($\alpha = 1$) is relatively small. In contrast, on ConCon-Chi, using only the context or the concept causes a sharp drop in performance, highlighting that to achieve good performance on it strong concept-context composition capabilities are required.

3.3.2 Novel Concepts Performance

I tested several state-of-the-art personalized TIR algorithms on ConCon-Chi and compared their performance on chimeric and common concepts. The results show that all methods perform significantly worse on chimeric concepts. To understand whether the low performances on chimeric concepts were caused by poor concept recognition or poor concept-context compositionality, I measured the F1 score for concept retrieval, i.e., an image is considered a positive for a certain query if it contains the specified concept regardless of the context. The performance on concept retrieval is similar on chimeric and common concepts. This indicates that the main challenge posed by chimeric concepts is not recognizing them but rather composing them with different contexts.

Chapter 4

KDE-Based Trajectory Selection

My contribution in the field of robot learning focused on improving the performance of implicit behavior cloning policies through test-time computation and resulted in the publication *KDPE: A Kernel Density Estimation Strategy for Diffusion Policy Trajectory Selection* (Rosasco et al., 2025). The full publication is provided in Section 8.3.2.

Diffusion Policy enables the modeling of multimodal behaviors from human demonstrations. However, due to the stochastic nature of the denoising process, a single sampled trajectory at inference time might be an outlier with respect to the dominant modes in the training data, potentially leading the robot out-of-distribution. To mitigate this issue, KDPE generates multiple trajectories in parallel and uses Kernel Density Estimation (KDE), equipped with a manifold-aware kernel, to estimate the probability density of the resulting actions. By selecting the trajectory with the highest density, the method prioritizes actions that are more representative, improving the performance of the policy during closed-loop control.

4.1 Background

Robot learning typically refers to the research field concerned with the development and training of embodied policies for robot control. While the term is broad, in this thesis I will use it to refer to task-oriented, end-to-end models that map visual observations to robot actions. The two main paradigms used to train robot learning models are Reinforcement Learning (RL) and Behavior Cloning (BC). While RL has achieved impressive performance in a wide variety of domains (Degrave et al., 2022; Silver et al., 2016), it typically requires substantial interaction data, and in the case of vision-based policies, sim-to-real transfer remains challenging. For these reasons, in recent years, BC became widely adopted for the

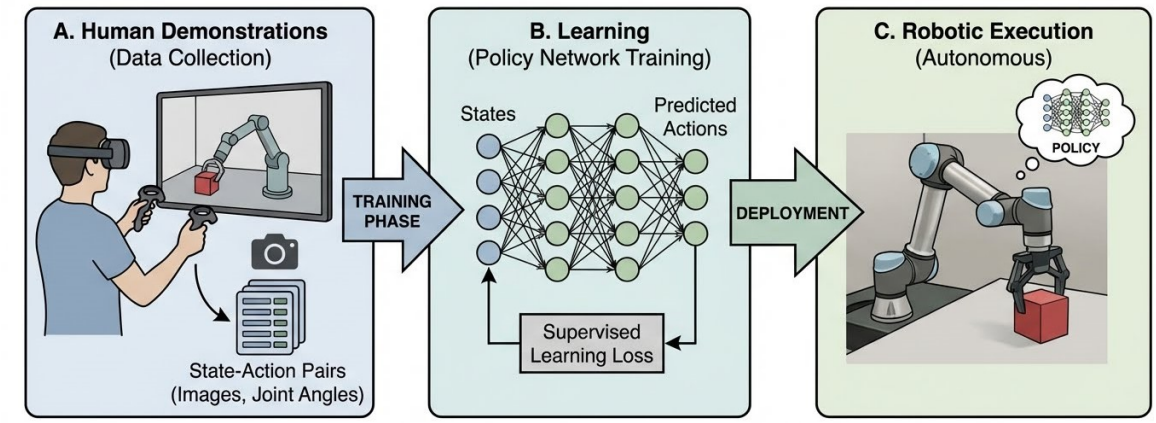


Figure 4.1 Behavior Cloning pipeline. A dataset of state-action pairs is collected by recording the robot sensors while it is being controlled to perform a task. A policy is then trained via supervised learning to predict the action given the state. At inference time, the trained policy is executed in closed-loop on the robot to perform the task.

training of visuomotor policies. In this section I will outline some general concepts related to BC and some more specific topics used by my proposed method in Chapter 4.

4.1.1 Behavior Cloning

Behavior Cloning (BC) is a learning paradigm often used in robotics which consists of training a policy $\pi_\theta(a|s)$ through supervised learning by setting as input an observation of the environment s (e.g., images and proprioception) and as the target the action a. To train a policy for a given task, a set of demonstrations is collected by recording the robot's sensor data while an operator teleoperates it to perform the task. These demonstrations are then reorganized into a dataset of state-action pairs that will be used to train the policy. Once the policy has been trained, it can be run in closed-loop on the robot to perform the task.

The datasets used to train BC policies are usually collected via human teleoperation posing a significant challenge as human-collected data can be noisy and present multimodal behaviors. In the context of behavior cloning, multimodality refers to the presence of different actions associated with similar states. A model trained on multimodal data with standard MSE loss learns to predict the average action given a certain state, leading to poor performance. To overcome this problem, recent works have proposed to use generative models to learn the conditional action distribution and then sample from it at inference time (Florence et al., 2022; Shafiullah et al., 2022). A particularly successful method that adopts this approach is Diffusion Policy (Chi et al., 2023).

4.1.2 Diffusion Policy

Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020) are a class of generative models that learn to generate data by reversing a gradual noising process. While Diffusion Models were initially studied in the context of image generation (Rombach et al., 2022; Song et al., 2021), they quickly found applications in multiple fields such as robotic control. Diffusion Policy (DP) adapts DDPMs to behavior cloning for robotic manipulation tasks. DP, given an observation of the environment, outputs a trajectory of actions of shape $T \times D$, where T is the number of time-steps and D is the action dimensionality. These trajectories are generated through a denoising process that progressively refines Gaussian noise by iteratively subtracting a learned gradient field. This process allows the policy to model multimodal trajectory distributions, which is key for learning from human-collected datasets. The denoising process to generate a trajectory is defined as:

$$\mathbf{A}^{k-1} = \alpha(\mathbf{A}^k - \gamma\varepsilon_\theta(\mathbf{A}^k, k) + \mathcal{N}(0, \sigma^2 I)), \tag{4.1}$$

where ε_θ is the noise prediction network with parameters θ and $\mathbf{A}^k$ is a noisy sample going through a denoising step. This process is repeated for K steps, starting with $\mathbf{A}^K$ as randomly sampled Gaussian noise, to output the trajectory $\mathbf{A}^0$. During training, a trajectory is sampled from the dataset and is perturbed by adding the appropriate amount of noise corresponding to a random denoising step k. The noise prediction network ε_θ takes as input the noisy sample and is optimized to predict the noise ε^k that has been added to the ground-truth trajectory $\mathbf{A}^0$. The loss function used to train the model is defined as

$$\mathscr{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathbf{A}^0 + \varepsilon^k, k)). \tag{4.2}$$

4.1.3 Kernel Density Estimation

KDE (Parzen, 1962) is a technique used to estimate the probability density function (PDF) of a probability distribution based on samples drawn from such distribution. Given a set of N IID samples $\{\mathbf{x}_i\}_{i=1}^N$ drawn from a distribution and a kernel function k, KDE is computed by centering the kernel function on each sample and summing the contributions of each function. Formally, we can write the kernel density estimator as:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_H(\mathbf{x} - \mathbf{x}_i), \tag{4.3}$$

where K_H is a kernel function with bandwidth H.

4.2 Method

4.2.1 Trajectory Selection

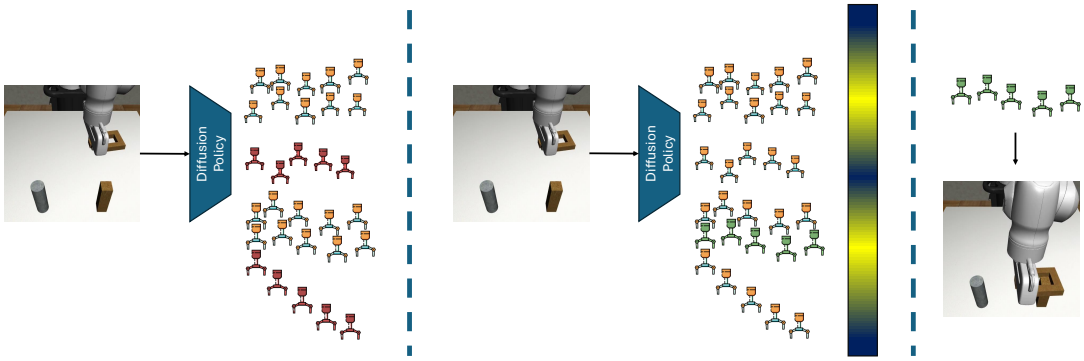


Figure 4.2 (Left) Trajectories sampled from Diffusion Policy can occasionally be outliers. This becomes apparent if a large enough population of trajectories is sampled from the policy. (Center) KDPE assigns a density to each trajectory computing KDE on every last action. (Right) The trajectory associated to the action with highest density is executed.

KDPE is a training-free technique that consists in applying a *best-of-N* strategy to the output of Diffusion Policy. At each control step, given the current observation, a population of candidate trajectories is sampled from the policy. While each trajectory is a short sequence of Cartesian actions, KDPE only considers the last action of each trajectory. Then, instead of selecting one of these trajectories uniformly at random, KDPE fits a KDE to the sampled actions and assigns to each candidate the estimated density (Figure 4.2). Finally, the trajectory associated to the highest density is executed on the robot.

4.2.2 Manifold-Aware KDE on Robot Actions

To compute densities, KDPE uses KDE with a kernel defined over the action space. The actions considered in this work include end-effector Cartesian position, end-effector orientation, and gripper aperture. While positions and gripper states are linear, rotations live on a non-Euclidean space so treating them as ordinary vectors with a Euclidean distance produces inconsistent densities.

KDPE uses a manifold-aware kernel that computes the difference between two orientations using the geodesic distance. This rotational distance is then combined with standard

Euclidean distances for position and gripper aperture. The kernel used in KDPE is an adaptation of the probability distribution presented in Barfoot and Furgale (2014) extended to include the gripper state.

Three separate bandwidths regulate the relative importance of position, orientation and gripper state.

4.3 Experimental Validation

4.3.1 Simulation Experiments

I evaluated KDPE against the DP baseline on seven simulated tasks from two benchmarks: *Lift*, *Can*, *Square*, and *ToolHang*, from RoboMimic (Mandlekar et al., 2021) and *Coffee*, *Stack Three*, and *Three Piece Assembly*, from MimicGen (Mandlekar et al., 2023). For each evaluation I fixed the environment resets and noise schedule seeds and ran 100 rollouts per task. For certain tasks, two different datasets are provided in RoboMimic: one collected by proficient human operators (*ph*) and one by mixed-quality operators (*mh*). I trained the policy on both datasets and report the results in both settings. KDPE improves the average success rate over DP for both CNN-based and Transformer-based backbones. The performance gain is stronger in settings where outliers matter the most: higher-precision tasks (e.g. *ToolHang*) and datasets collected with lower-quality demonstrations (*mh*).

4.3.2 Real-robot Evaluation

Finally, I tested KDPE on a Franka Emika Panda manipulator on four real-world tasks: *PickPlush*, a tabletop grasp-and-lift task; *PickSponge*, a domain shifted version of *PickPlush* that uses the same training data; *CubeSort*, a multi-step and highly multimodal pick-and-place task; and *CoffeeMaking*, a long-horizon task requiring precise execution. Results show consistent improvements over the DP baseline. Beyond success rates, a recurring qualitative difference is that KDPE tends to avoid abrupt changes in end-effector and gripper motion, which matches the intuition behind discarding low-density (outlier) trajectories.

In terms of computation, KDPE adds only a modest overhead, keeping the approach suitable for closed-loop visuomotor control, where action selection must happen at every time-step.

Chapter 5

Robotic Demonstration

In addition to the scientific publications above, I contributed to the development of a human-robot interaction demo on the ergoCub humanoid robot. While this work did not lead to a publication (as it focused primarily on system integration rather than methodological novelty), it required integrating perception and control components into a reliable system.

In particular, I contributed to: (i) inter-module communication and data flow between perception, planning, and control components, (ii) the design and implementation of a behavior tree to coordinate the modules, (iii) training and deploying a lightweight segmentation model tailored to the demo setup, and (iv) a visual servoing controller for the head.

The components were orchestrated by the behavior tree (Figure 5.1) to interpret the interacting human's behavior and trigger pre-scripted responses. Human behavior was inferred through a combination of few-shot action recognition (Berti et al., 2022) and object segmentation. The system components ran in parallel on multiple machines, with communication between them implemented through the YARP middleware (Metta et al., 2006). In particular, lightweight modules such as the controllers were executed on ergoCub on-board computers to reduce latency, while deep learning based modules ran on an external GPU-equipped laptop.

In practice, the robot reacted to a set of human actions (e.g. waving, shaking hands), grasped a box when it was handed to it, and passed it back when required. In parallel, the head was controlled through visual servoing to keep either the human or the object in the center of the frame. Finally, when triggered by the corresponding action, the robot started walking to approach the human. The use of a behavior tree for high-level control made it straightforward to define conditions for switching between behaviors and to specify fallback actions, helping the robot react more promptly to changes in the interaction.

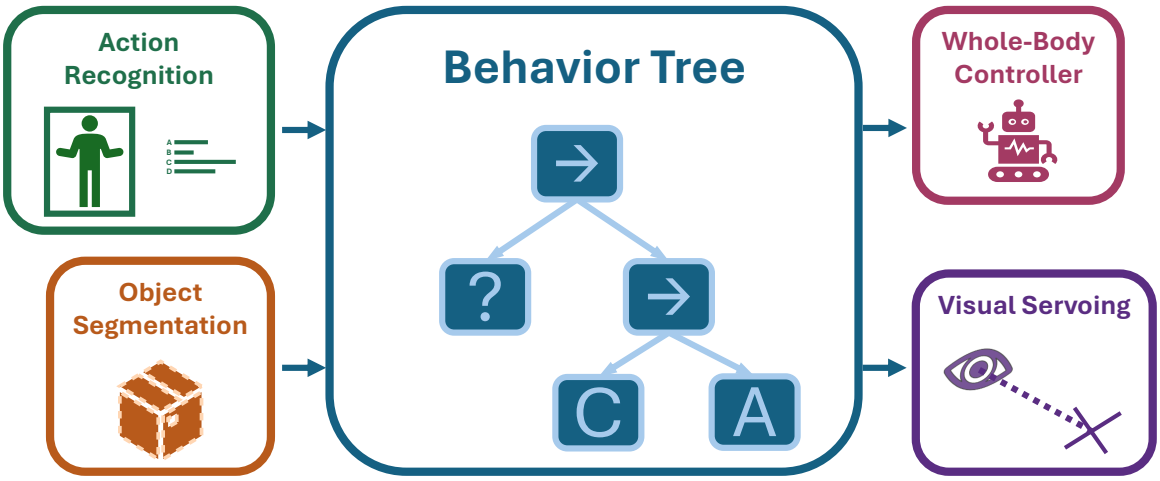


Figure 5.1 Block diagram of the Human-Robot interaction demo implemented on the ergoCub robot. The behavior tree continuously orchestrates different modules to exhibit a reactive behavior. Perception-related components (i.e. action recognition and object segmentation) are used by the behavior tree to define conditions under which specific actions are executed through control components (i.e. whole-body controller and visual servoing)

I performed this demonstration multiple times, both in the laboratory for external visitors and at external events such as ICRA 2023, ECCV 2024, and Festival del Lavoro 2024.

Part III

Published Research

Chapter 6

Towards Confidence-guided Shape Completion for Robotic Applications

Andrea Rosasco, Stefano Berti, Fabrizio Bottarel,
Michele Colledanchise, and Lorenzo Natale

Abstract

Many robotic tasks involving some form of 3D visual perception greatly benefit from a complete knowledge of the working environment. However, robots often have to tackle unstructured environments and their onboard visual sensors can only provide incomplete information due to limited workspaces, clutter or object self-occlusion. In recent years, deep learning architectures for shape completion have begun taking traction as effective means of inferring a complete 3D object representation from partial visual data. Nevertheless, most of the existing state-of-the-art approaches provide a fixed output resolution in the form of voxel grids, strictly related to the size of the neural network output stage. While this is enough for some tasks, e.g. obstacle avoidance in navigation, grasping and manipulation require finer resolutions and simply scaling up the neural network outputs is computationally expensive. In this paper, we address this limitation by proposing an object shape completion method based on an implicit 3D representation providing a confidence value for each reconstructed point. As a second contribution, we propose a gradient-based method for efficiently sampling such implicit function at an arbitrary resolution, tunable at inference time. We experimentally validate our approach by comparing reconstructed shape with ground truth, and by deploying our shape completion algorithm in a robotic grasping pipeline. In both cases, we compare results with a state-of-the-art shape completion approach. The code is available at https://github.com/andrearosasco/hyperpcr

6.1 Introduction

Nowadays, 3D environment perception for robotics is as affordable and available as it ever was. In some contexts, the environment the robot is operating in is completely known by design, e.g. industrial work cells, or can be thoroughly explored, simplifying tasks such as navigation or object manipulation. In many contexts, however, robots cannot rely on full knowledge of the working environment due to a limited workspace, unreachable viewpoints, or clutter, and only partial 3D representations are available. In the case of object grasping and manipulation, for instance, a robot might only be allowed partial views of a target object with unknown geometry. In this scenario, pose detection and classical grasp planning techniques cannot be employed. While it may still be possible to grasp and manipulate the object using partial 3D data, in latest years methods aiming at shape reconstruction (or *shape completion*) using deep learning approaches have proven to be a promising research direction.

One of the first successful approaches to shape completion consists in extending Convolutional Neural Networks (CNNs) to 3D shapes. While these methods manage to achieve interesting results, they are constrained to a voxelized representation of their input and output shape. This results in a trade-off with the network complexity scaling cubically with the resolution. After the introduction of input layers capable of processing unstructured set of points, various works started to operate directly on point clouds. These approaches have the benefits of relying on an input format both lightweight and easy to extract from the data provided by 3D sensors. Despite the representative power of point clouds, methods that use them as output representation often suffer from fixed output resolution, as the number of generated points depends on the output layer dimension. Learning an implicit representation yields different advantages such as reducing the number of parameters of the model and allowing for an arbitrary resolution of the output despite having a fixed-size implicit function.

To this end, we propose a framework that achieves arbitrary output resolution. It consists in generating an implicit function that captures a continuous model of the output space. We structured our model as a HyperNetwork (Ha et al., 2017), a particular neural network architecture where a primary network generates the weights for a secondary network used to perform the task. The structure allows to reduce the size of the model in terms of learnable parameters and implement a fine-tuning routine to better exploit the data available at evaluation time.

The main contributions of this work are:

- a novel HyperNetwork-based architecture for shape completion using a transformer encoder

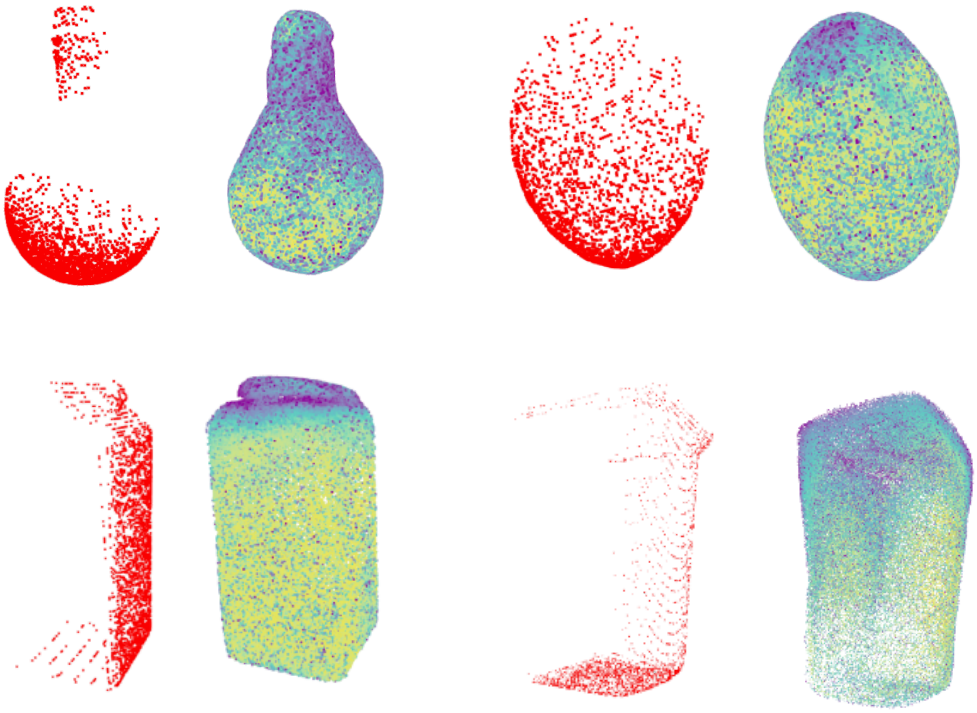


Figure 6.1 Partial point clouds and reconstructions of four objects. The color scheme of the reconstruction indicates the uncertainty of the model on each point (brighter means higher certainty). Particularly the model tends to assign higher probabilities to points near the visible part of the object. This could help in grasp evaluation.

- a gradient-based sampling algorithm to reconstruct the point cloud from the implicit representation.

We experimentally validate our proposed approach in two stages. First, we compare reconstructed shapes with ground truth using a popular object dataset. Second, we integrate our shape completion method in a state-of-the-art visual-based grasp planning pipeline. In both experiments, we compare results with a state-of-the-art shape completion deep learning architecture and provide guidelines to reproduce our experiments.

The rest of the paper is organized as follows. Section 6.2 briefly outlines works related to object modeling and shape completion. Section 6.3 summarizes some background information useful in the understanding of this work. Section 6.4 formalizes the shape reconstruction problem and Section 6.5 details the proposed solution. Section 6.6 explains our experimental methodology and Section 6.7 closes off the paper by summarizing our results and proposing a future work direction.

6.2 Related Work

Existing shape completion algorithms can be classified based on the representation they use for their input and output data. Two popular shape representations are point clouds and voxels.

6.2.1 Voxel-based

Voxels are the extension of pixels to the three-dimensional space. While they can be represented with different data structures, the one commonly used for learning-based shape completion is that of an occupancy grid. Occupancy grid representation allows a straightforward adaptation of convolutional neural networks to the 3D space (Choy et al., 2016). This approach has the advantage of both relying on a structured representation of the object and on a widely tested architecture (i.e., CNNs), but suffers from high computational cost. Since the number of operations scales cubically with the resolution of the occupancy grid in the current implementations, we cannot use them to process and generate shapes with fine-grained details.

Although some voxel-based algorithms can manage resolutions up to 128^3 (Wu et al., 2018), they require a considerable amount of memory constraining the algorithm to shallow architectures.

6.2.2 Point-based

Point clouds are often used as a representation for 3D data, given their ability to represent fine details. Their sparse nature, however, precludes direct applicability of convolution operators. The introduction of input layers able to efficiently process point clouds made them a viable representation. PointNet (Qi et al., 2017) was one of the first methods to work directly on point clouds. Then, an increasing number of works started to use them as input and output representations. Since these methods regress the output points coordinates they are usually unable to output a confidence value associated to each point.

6.2.3 Shape Completion for Grasping

Grasping algorithms can be categorized into model-based and model-free, depending on whether or not specific knowledge about the object (e.g., CAD model or previously scanned model (Bormann et al., 2019)) is required to solve the considered task. Model-based algorithms (Tremblay et al., 2018) usually include a pose estimation step to match the model of

the object to the real object point cloud. While this usually leads to precise grasps, it requires the object to be known a priori. This assumption is oftentimes unrealistic.

Model-free approaches (Mahler et al., 2016) instead, do not need prior knowledge of the input object and operate just on its partial representation obtained from the robot sensors. While some grasping algorithms can directly operate on partial views, shape completion can help to increase the probability of finding a good grasp.

Some of the first algorithms that used completion were guided by the geometry of the object and attempted to reconstruct the occluded part exploiting symmetry (Bohg et al., 2011) or by the use of heuristics (Schnabel et al., 2009). The work of Varley et al. (2017) was one of the first to pioneer deep learning-based shape completion applied to robotic grasping. Following their work, Lundell et al. (2019) developed a voxel-based architecture that uses Monte Carlo dropout to measure the uncertainty of the shape. More recent work uses implicit functions as an output representation in the form of signed distance functions (Van der Merwe et al., 2020) or mapping functions (Yang et al., 2021).

6.3 Background

6.3.1 HyperNetwork

HyperNetwork (Ha et al., 2017) refers to a particular neural network architecture composed of a primary network f and a secondary network g, where f is responsible for generating the weights of g and g generates the final output. This network architecture proves particularly efficient in terms of parameters as the only learnable weights are the ones of the primary networks and it also directly learns an implicit function of the output space. Thanks to their properties, Hyper-Networks were applied to image-to-image translation (Jia et al., 2016), neural architecture search (NAS) (Zhang et al., 2019) and shape reconstruction (Littwin and Wolf, 2019).

6.3.2 Transformer

Transformers (Vaswani et al., 2017) are an attention-based architecture first introduced as a solution for Natural Language Processing (NLP) tasks. The fundamental mechanism of the transformer is self-attention. Attention consists in measuring the similarity of a query embedding against a set of key embedding through dot product. This produces a list of similarity scores that is used to take a weighted sum of the value embeddings. When this procedure is applied for each element of a sequence against all the other elements, it takes

the name of self-attention. Transformers are also based on an encoder-decoder structure that makes them perfect for sequence-to-sequence tasks (e.g. machine translation). Despite that, both the attention-based encoder and decoder of the transformer have been used by themselves with great success (Devlin et al., 2019). While the ability to pick up on long sequences of elements automatically made the transformer a good fit for language tasks, it was not immediately clear how to exploit such architectures in other tasks. The publication of the Vision Transformer (ViT) (Kolesnikov et al., 2021) showed how the architecture could attain good results in image recognition if properly designed. After ViT, an increasing number of publications (Liu et al., 2021a) successfully applied transformers to computer vision tasks achieving state-of-the-art results.

6.3.3 Geometry-aware Transformer

Recently, after their first successful application to computer vision, the transformers have been successfully applied to shape reconstruction. PoinTr (Yu et al., 2021) is a transformer-based architecture that uses a geometric-aware attention layer to introduce an inductive bias in the model.

Since the transformer computational complexity scales quadratically with the input sequence length, the point cloud cannot be directly passed as input to the transformer but is pre-processed to generate a shorter sequence of embedding $\mathscr{F} = \{\mathscr{F}_0, ..., \mathscr{F}_N\}$ representing the point cloud. Attention between the inputs' embeddings is computed using the standard self-attention equation:

$$softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{6.1}$$

At the same time, following the work of Yu et al. (2021), the edge between each proxy point and the n closest proxy points are passed through a linear layer generating the edge embeddings:

$$E_{i,j} = ReLU(\theta \cdot (\mathscr{F}_j - \mathscr{F}_i) + \phi \cdot \mathscr{F}_i). \tag{6.2}$$

Then the n edge embedding gets aggregated through max-pooling. The newly computed geometric embeddings have the same dimensionality as the point proxies, but they encode more information about their surroundings. Then, the output of the self-attention layer is concatenated to one of the geometric layers and mapped to its original dimension to generate the output.

Since PoinTr treats the shape completion task as a sequence-to-sequence problem on point proxies, the output of the transformer is a sequence of point proxies $\mathscr{H} = \{\mathscr{H}_0, ..., \mathscr{H}_M\}$

describing the full shape. To reduce the computational complexity, M has to be low. Therefore, to generate a fine-grained point cloud, a multi-scale approach is applied. In particular, similarly to PCN(Yuan et al., 2018), the point proxies are passed to a FoldingNet (Yang et al., 2018) that deforms a 2D grid of points to generate the complete point cloud.

6.3.4 Grasp Pose Detection (GPD)

GPD (ten Pas et al., 2017b) is a deep learning method taking point clouds as input in order to produce 6D grasp candidates with no a priori knowledge of the target CAD model. It works by evenly sampling candidates on the point cloud surface and filtering them according to geometric criteria. Each raw candidate is processed through a CNN to obtain a measure of quality, indicating how much each candidate resembles a frictionless antipodal grasp. To convert such input into a CNN-compatible representation, the portion of the point cloud enclosed in the gripper volume is voxelized. Three images from the perspective of each axis of the hand reference frame are then computed: the heightmap of the occupied voxels, the heightmap of the unoccupied voxels and the surface normal map. This results in a 9-channel image that can be processed by a CNN.

6.4 Problem Formulation

Given an object O, let $Y \triangleq PC_{3D}(O) \in \mathbb{R}^{n\times 3}$ be a complete point cloud of O, and $X \triangleq PC_{2.5D}(O) \in \mathbb{R}^{n\times 3}$ a partial self-occluded point cloud. Let $m : \mathbb{R}^{n\times 3}, \mathbb{R}^{n\times 3} \rightarrow \mathbb{R}$ be a metric that gives a measure of the dissimilarity between the two point clouds and δ be a confidence bound. We define the shape completion problem as the problem of finding function $f : \mathbb{R}^{n\times 3} \rightarrow \mathbb{R}^{n\times 3}$ such that $f(X) = \widetilde{Y}$ and $m(Y, \widetilde{Y}) \leq \delta$

6.5 Proposed Solution

In this section we introduce our architecture to estimate a 3D reconstruction of an object from its 2.5D point cloud, in the form of an implicit function of the 3D space. To achieve arbitrary resolution and obtain a measure of uncertainty of each output point, we define an implicit shape representation that can be sampled on real coordinates whose values give us information about the complete shape of the object. We adopt a hyper-network structure where the weights ϕ of an implicit function g_ϕ describing the complete shape of the object are generated by the backbone f. Using this type of architecture, we manage to reduce

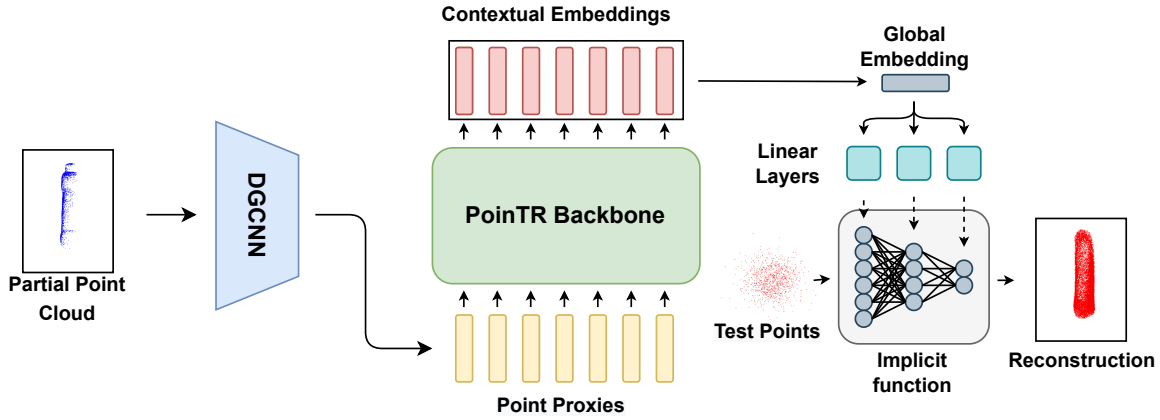


Figure 6.2 Overview of our shape completion architecture. The backbone computes the contextual embeddings of the input point proxies. Then, we pass the embeddings through a max-pooling layer to obtain the global embedding. Finally, we feed the global embedding to different linear layers to generate weights, biases and scales of the implicit function.

the number of parameters of the model while obtaining good performances on the shape completion task. We define g_ϕ as a parametric occupancy function

$$g_\phi : \mathbb{R}^3 \to [0, 1] \tag{6.3}$$

that returns the model's confidence that a query point $p \in \mathbb{R}^3$ lies on the object surface. Since the ground truth is provided as a finite point cloud Y, we define a binary target function with a tolerance $\varepsilon > 0$:

$$g_Y(p) = \begin{cases} 1, & \text{if } d(p,Y) \leq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

where $d(p,Y) \triangleq \min_{y \in Y} \|p - y\|_2$ and $p = (x,y,z)$ is a tridimensional point.

To condition g_ϕ on past knowledge extracted from a dataset, we introduce an encoder module f_θ that takes as input a partial point cloud $X \in \mathbb{R}^{n \times 3}$ and returns the weights ϕ that parameterize the implicit function g_ϕ.

6.5.1 Implicit Function

We implemented the implicit function g_ϕ as a multi-layer perceptron whose weights are generated by a backbone every time a partial input is processed.

Following previous work Littwin and Wolf (2019) we generate three sets of weights for each layer l:

$$\phi_l = \{\phi_l^W, \phi_l^b, \phi_l^s\}$$

respectively the weight, bias and scale parameters. Given a specific layer l, its input x_l, and the parameters ϕ_l, the output of the layer is computed as follows:

$$y_l = ((\phi_l^W x_l) \cdot \phi_l^s) + \phi_l^b. \tag{6.5}$$

In particular, we use a multi-layer perception with two hidden layers of dimension 32 with Leaky ReLU activation functions set to 0.2.

6.5.2 Backbone

As the backbone of our model, we use the encoder of the PoinTr transformer introduced in Section 6.3 with one geometric layer in the first transformer block. Following their example, we compute the point proxies of the input point cloud using a hierarchical version of Dynamic Graph CNN (Wang et al., 2019). This input layer uses EdgeConv operator to reduce the full point cloud to a fixed number of embeddings that we call point proxies, each representing a different part of the input. After computing the N point proxies, we input them to the transformer as a sequence and get a contextual embedding for each of them. Thanks to self-attention, these embeddings represent each point proxy in relation to the other ones. To get a global representation of the point cloud, we apply max pooling to the embeddings and obtain the global embedding e. The global embedding is then processed by a set of linear layers to generate the weights ϕ for each layer of the implicit function.

6.5.3 Implicit Function Input

During training, in addition to the partial point cloud taken as input by the backbone, we create a batch of points spanning the output space to train the implicit function. We sample half of the points from the ground truth point cloud and label them as positive examples. The other half is composed by a 10% of points uniformly sampled across the output space and a 40% obtained by adding Gaussian noise to points from the ground truth. This is motivated by the fact that points sampled near the objects' boundaries hold more information about the shape of the object and facilitate training. To assign a label to the perturbed and the uniformly sampled points, we set $y = g_Y(p)$, i.e., we check their distance to the points on the ground truth. If they are close enough to it they are assigned a positive label.

We then feed the batch of points to the implicit function that returns, for each point, a value between 0 and 1 using a sigmoid activation function. We compute the binary cross entropy loss between the predictions and the labels and backpropagate the gradient to the backbone that updates its parameters. This results in learning a model that is able to map a partial point cloud to a function representing its respective complete point cloud.

6.5.4 Point Cloud Sampling

Once the backbone has produced an implicit function representation of the complete shape, we might need to sample a point cloud from it. The naive approach would be to create a grid of points and test each of them through the implicit function. The issue with this method is that just the small fraction of points that intersect the implicit surface will contribute to forming the point cloud. This problem worsens if we increase the classification threshold to gather points with a higher probability of being part of the object. Furthermore, increasing the sampling resolution increases the number of points to be tested cubically, increasing the risk of running out of memory. To solve this problem, we developed an algorithm that uses the first-order information associated with each point to iteratively reconstruct the input point cloud. Our algorithm steps are shown in Algorithm 1. Note that in our implementation we omit the for loop in Line 4 as we operate on matrices of points. We initialize a list of points $\mathscr{L}$ by randomly sampling from a Gaussian distribution centred in 0 and with standard deviation 0.1 (Line 2). Then, for each point, we get the output of the implicit function and add it to the output if its probability is higher than the threshold (Lines 4 to 9). After that, we measure the loss between the probability assigned from the implicit function and probability 1. Through back-propagation we compute the gradient of the loss function with respect to the input point and update it through gradient descent (Line 10). Finally, we update the point in $\mathscr{L}$. In short, the points in $\mathscr{L}$ move toward the surface described by the implicit function g_ϕ and every time one of the points crosses the threshold we copy it and add it to our output point cloud.

6.6 Experimental Validation

We run two sets of experiments: one where we evaluate the performance of our shape completion algorithm and compare it with the state of the art; and one where we give the reconstructed shape to a grasping algorithm and measure the grasp success rate. We compare our shape completion results with the one presented by Varley (Varley et al., 2017) (V) and

Algorithm 1 Point Cloud Sampling

Inputs: g_ϕ: implicit function; N: initial sample size; η: step size
Inputs: M: number of output points; τ: threshold; ℓ: loss function

$reconstruction \leftarrow$empty list
$points \leftarrow$list of N 3D points sampled from $\mathcal{N}(0, 0.1)$
while $reconstruction.Size() < M$ **do**
  for $i = 1$ ***to*** $points.Size()$ **do**
    $x = points[i]$
    $\tilde{y} = g_\phi(x)$
    if $\tilde{y} > \tau$ **then**
      $reconstruction.insert(x)$
    end if
    $points[i] = x - \eta \nabla_x \ell(\tilde{y}, 1)$
  end for
end while
Output: $reconstruction$

the one presented by Lundell (Lundell et al., 2019) (USN). Then we choose GPD as our grasp planner and compare our grasping pipeline with one composed of USN shape completion and GPD. Both V and USN use a 3D CNN to generate a $40 \times 40 \times 40$ voxel grid of the complete shape. Then, to convert their output to a representation more suitable for grasping, they apply a surface reconstruction algorithm introduced in (Varley et al., 2017) to generate a mesh. The CNN used by Varley applies three convolutions to the input voxel grid followed by two dense layers. The output of the dense layer is then reshaped in the final voxel grid. USN instead adopt an encoder-decoder architecture with skip connections that reconstruct the final shape through deconvolution. Furthermore, USN applies dropout at inference time to generate different reconstructions of the same input. The reconstructions are then averaged to generate the complete shape.

6.6.1 Dataset

We trained our model on self-occluded point clouds generated from meshes from the Columbia Grasp Database (Goldfeder et al., 2009) and YCB (Calli et al., 2015). Following the procedure used in Lundell et al. (2019) we split the test data in

- Holdout Views: views of objects that have already been observed during training, but from different angles.
- Holdout models views: views of objects never observed during training.

We use the same splits provided by V. This resulted in about 200k training samples, 60k holdout views and 100k holdout models views. We also created a small validation set containing all the views of 4 models from the training set to be used for early stopping and hyper-parameter selection.

The partial views of the objects were generated by loading the models in Gazebo and rendering their depth image. Each object was rotated around the three axes to generate a total of 726 partial views.

Figure 6.3 Official YCB names following the numbering order and skipping the object number 8 (not official YCB object): airplane toy, power drill, mustard container, glass cleaner, box of sugar, bleach cleanser, chips can, pitcher. We consider object number 10 as the pitcher without its lid on.

6.6.2 Training Procedure

We trained our model on an NVIDIA V100 for 60 epochs. The model was updated with Adam optimizer with a learning rate of $1e^{-4}$ and a mini-batch size of 32. We set the encoder depth to 4. We use 6 attention heads and an input embedding dimension of 384. For the grasping experiments, we perturbed the input point cloud with noise to make it more robust to the input provided by the depth camera.

6.6.3 Shape Completion

To evaluate their algorithms, Varley and Lundell, generated the test data by sampling 50 views from the training set, 50 views from the holdout views set and 50 views from the holdout models views set. We assume that they did not test on the full dataset due to the meshing process being too computationally expensive. Unfortunately, we did not manage to use the surface reconstruction algorithm introduced by V and used by USN to replicate their results. For this reason, we report the results as indicated in their work. Testing only

	V	USN	Ours
Training Views	0.6205	0.6446	**0.6712**
Holdout Views	0.6143	0.6389	**0.6667**
Holdout Models	0.5632	0.5651	**0.6023**

Table 6.1 Jaccard Similarity results. (Varley et al., 2017) and (Lundell et al., 2019) reported their result on a random sample of 50 examples per split. For our algorithm we report the results on all the samples in each of the three splits.

our architecture on randomly sampled subsets of 50 of the three splits would not provide a good comparison: two different subset of dimension 50 from a set of 100k elements can lead to very different result. For this reason, we report the results of our algorithm computed on the whole splits.

We assess the shape completion performance of our algorithm against USN by measuring the Jaccard Similarity between the ground truth and our reconstruction:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{6.6}$$

In particular, we first convert our prediction and the corresponding ground truth to a voxel grid of size 40^3 and then check the number of voxel occupied by their union and by their intersection. To generate our output point cloud we used the sampling technique showed in Section 6.5. We set the generated number of points to $1e^5$, then we downsample the point cloud with farthest point sampling to 16384 to obtain a more uniform reconstruction.

The average results of our method and USN are shown in Table 6.1. We can see that our algorithm is able to score a higher Jaccard similarity on the three splits. To attain our results we selected on the validation set the best parameters for our sampling algorithm: 20 iterations, a threshold of 0.85 and an update step of 0.1.

6.6.4 Grasping Experiments

Like Lundell et al. (2019), for the grasping experiments we used 8 holdout YCB objects plus the power drill box. The objects are shown in Figure 6.3. We also tried to grasp the pitcher (number 9) without its lid on and considered it object number 10. The robotic platform consists in a Franka Emika Panda arm equipped with a Robotiq 2F-85 gripper and a RealSense D415. The setup is shown in Figure 6.4. Following the same protocol as Lundell et al. (2019), we attempted a total of 10 grasps per object on five different orientations (0°, 72°, 144°, 216°, and 288°) for a total of 100 grasps per method. Differently from Lundell

Figure 6.4 Our robotic grasping set-up.

et al. (2019), our camera is hand-mounted and the 2F-85 gripper features a much smaller graspable volume with respect to the Barrett hand used in that work. In order to fairly compare the effectiveness of USN against our approach in a grasping scenario, we generate completed point clouds using both algorithms and feed them in turn to the GPD planner. A grasp is considered successful if the robot is able to lift the object off the table and hold it for 5 seconds.

The results on each test object are reported in Figure 6.6. It is interesting to notice how, on average, the success rates achieved by USN in our experiments are higher than the results reported in their work. This is most probably due to the use of GPD as a grasping algorithm and its robustness to noisy inputs.

The only object on which USN performances are considerably lower than the one obtained in their experiments is the pitcher. The reason is that the gripper they used, a Barrett hand, is big enough to grasp the pitcher without using its handle. Our gripper is considerably smaller and could grasp the pitcher just from its handle.

While our method struggles with some of the more complex objects, on average we perform better than USN. In particular, we score a total 75% success rate against USN 65%.

We observed that this is due to USN's tendency to reconstruct the objects just by increasing the volume around the partial observation. It seems that the predictions generated by USN prioritize precision over recall, generating more voxels close to the partial observation.

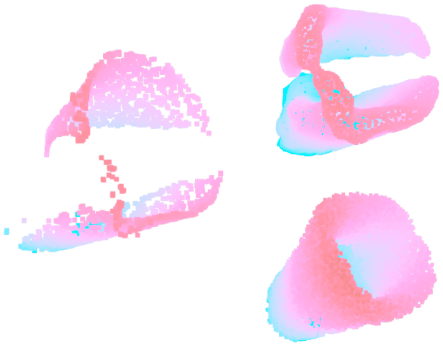


Figure 6.5 Left: partial point cloud of the pitcher without its lid. Top right: USN reconstruction. Bottom right: our reconstruction. Note the correspondence between the partial shape and USN reconstruction. The colors are just used to give a sense of perspective.

This means that several times USN reconstructions did not have enough depth, causing the predicted grasp to collide with the object. This can be noticed in Figure 6.5.

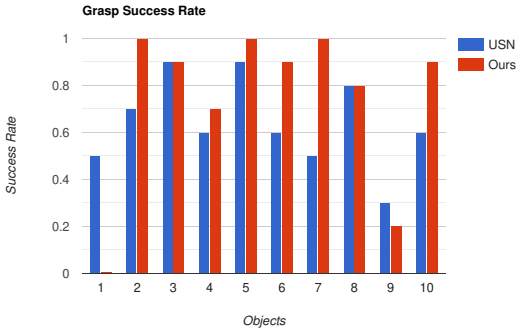


Figure 6.6 Comparison of the grasp success rate of our pipeline and USN pipeline on each object.

Our algorithm on the other end tends to favour recall, reconstructing the surface of the objects without sticking too close to the partial observation. As shown in Figure 6.1, while the area closer to the input point cloud has a higher probability, the whole shapes are correctly reconstructed.

The grasping success rate on object 1, the toy aeroplane, reflects the tendency of our algorithm to favour recall. Our method consistently filled the space between the upper and lower wing making it almost impossible to grasp. This is not surprising since with respect to the training data object 1 is out of distribution.

6.7 Conclusion

We presented a novel algorithm for shape completion based on a HyperNetwork-based architecture that uses the encoder of a transformer to generate the weights of an occupancy function. Having an implicit function as our output representation allows us to get a measure of confidence for each point of the output space. Moreover, despite the size of the occupancy function being fixed, we can use it to sample a point cloud at arbitrary resolution. We also introduced a technique to sample a dense point cloud of high confidence points from our implicit function, using the gradient to maximize the points' probability.

Quantitative experiments demonstrate superior performances of our model for the shape completion task on the Columbia Grasp Database. We tested on real hardware how the quality of the reconstruction affects the grasping success rate, showing that our reconstructions lead to better performance with respect to the previous state-of-the-art.

In future work, we would like to explore how the confidence measure provided by our model can be used to influence the output of a grasping algorithm, for example favouring grasps near high confidence areas of the reconstruction.

6.8 Limitations

The experiments presented in this work were designed to evaluate our shape completion method in the target robotic grasping setting. Given the scope of the work and the time constraints, some analyses, while valuable, are left to future work.

Gradient-based Sampling In Section 6.5.4 we motivate the introduction of our gradient-based sampling algorithm as an alternative to naive grid sampling. During experimentation we empirically observed that gradient-based sampling tends to produce more uniform point cloud reconstructions; however, a quantitative study of different sampling strategies would have provided a clearer assessment of its contribution.

Output representation While in this work we compare our implicit function-based architecture with voxel-based methods, a controlled study of how the choice of an output representation influences the downstream success rate for a fixed backbone could provide useful insight. Ultimately, our choice of output representation was motivated by its property of providing a fine-grained confidence measure that could be used in future work to condition active perception or grasp detection algorithms.

Comparisons The baseline methods included in the comparisons were representative of the current state of the art (2022) for shape completion for robotic grasping, and share an

evaluation framework that we also adopted. Shape completion is also widely studied in computer vision and the adoption and testing of methods typically introduced in this setting would strengthen the experimental validation of this work.

Chapter 7

ConCon-Chi: Concept-Context Chimera Benchmark for Personalized Vision-Language Tasks

Andrea Rosasco, Stefano Berti*, Giulia Pasquale*, Damiano Malafronte,*
Shogo Sato, Hiroyuki Segawa, Tetsugo Inada, Lorenzo Natale
* Equal contribution.

Abstract

While recent Vision-Language (VL) models excel at open-vocabulary tasks, it is unclear how to use them with specific or uncommon concepts. Personalized Text-to-Image Retrieval (TIR) or Generation (TIG) are recently introduced tasks that represent this challenge, where the VL model has to learn a concept from few images and respectively discriminate or generate images of the target concept in arbitrary contexts. We identify the ability to learn new meanings and their compositionality with known ones as two key properties of a personalized system. We show that the available benchmarks offer a limited validation of personalized textual concept learning from images with respect to the above properties and introduce ConCon-Chi as a benchmark for both personalized TIR and TIG, designed to fill this gap. We modelled the new-meaning concepts by crafting chimeric objects and formulating a large, varied set of contexts where we photographed each object. To promote the compositionality assessment of the learned concepts with known contexts, we combined different contexts with the same concept, and vice-versa. We carry out a thorough evaluation of state-of-the-art methods on the resulting dataset. Our study suggests that future work on personalized TIR and TIG methods should focus on the above key properties, and we propose principles and a

dataset for their performance assessment. Dataset: https://doi.org/10.48557/QJ1166 and code: https://github.com/hsp-iit/concon-chi_benchmark.

7.1 Introduction

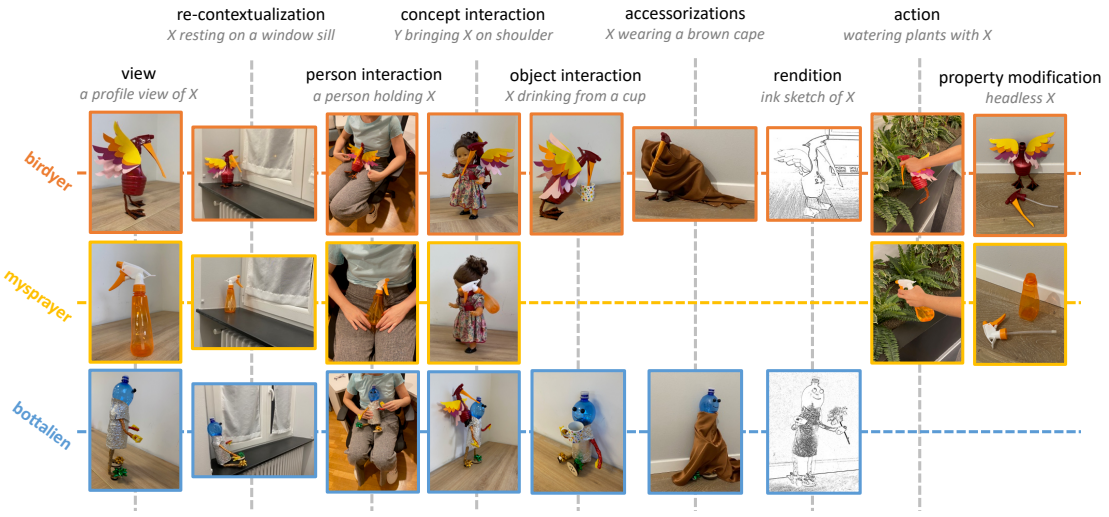


Figure 7.1 **Concept-context structure of the ConCon-Chi dataset.** Excerpt from the dataset structure (see Figure 7.4). Each row represents images of a concept (concept name in Color) and each column images of a context (context kind in Black and context in Grey). The caption for an image is formed by the composition of the context with the concept (replacing X/Y with the concept textual identifier).

Recent Vision-Language (VL) models for discriminative and generative tasks (Li et al., 2022a; Radford et al., 2021; Rombach et al., 2022) excel at associating textual descriptions with images. This resulted in a paradigm shift from closed to open-vocabulary versions of several computer vision tasks. In these settings, the model is not bound to a closed set of predetermined classes but can operate on free-form textual descriptions. However, it can be difficult for a user to formulate a description such that the VL model returns the expected output (Menon and Vondrick, 2023; Stretcu et al., 2023). The problem boils down to finding a text input whose encoding is close to the visual embedding of the target concept. Since this mapping is typically learned from Web-scale data, crafting effective descriptions for uncommon, novel or specific concepts, is challenging, thus hampering performance of VL models.

To represent this challenge, the "personalized" versions of Text-to-Image Retrieval (Cohen et al., 2022) (TIR) and Text-to-Image Generation (Gal et al., 2023a) (TIG) have been

recently proposed. The tasks consist of learning a user-specific *concept* using a few images and then performing retrieval (TIR) or generation (TIG) of such *concept* in a known *context*. Another similar task recently proposed is Zero-Shot Composed Image Retrieval (ZS-CIR) (Baldrati et al., 2023; Saito et al., 2023), the retrieval of a reference image modified according to a relative caption (Vo et al., 2019).

Due to lack of benchmarks for personalization tasks, many works proposed their method alongside a new dataset. However, most of these benchmarks lack two key properties necessary for a thorough evaluation of personalized concept learning: **novel concepts** and **compositional** structure.

Existing datasets use, as concepts, instances of common objects (e.g., in PerVL, clothing items from DeepFashion2 (Ge et al., 2019)). However, these allow to evaluate the learning of new words for known concepts (closely related to synonym matching, or instance identification), rather than new concepts. To simulate a realistic personalization setting, inspired by Lazaridou et al. (2017), we introduce chimeric concepts: objects created by the union of two unrelated existing concepts (see Figure 7.2).

The number and variability of contexts in which a concept appears is also underrepresented in personalized TIR/TIG benchmarks and this hampers the compositionality assessment of the newly learned concept with known contexts. Conversely, in a personalized TIR dataset multiple concepts should also appear in a same context, to prevent a method from attaining high performance by just attending to the context and disregarding the concept in the query ("context bias", pointed out also in ZS-CIR benchmarks (Baldrati et al., 2023; Saito et al., 2023)). To avoid both these problems, we formulated a large, varied, set of contexts for each concept, while also ensuring that each context can be composed with several concepts. As a result, the dataset has a concept-context matrix structure (Figure 7.1).

Our contributions are as follows:

- We highlight two key properties of personalized textual concept learning from images: learning new meanings and composing them with known ones; we propose the design of ConCon-Chi to model this problem.
- By evaluating on ConCon-Chi we show the limitations of state-of-the-art methods with respect to these properties.
- We release the dataset as a twofold benchmark for personalized TIR *and* TIG.

In the remainder of the paper, we compare related datasets in Section 7.2; we present ConCon-Chi in Section 7.3; we present our study for personalized TIR in Section 7.4 and for personalized TIG in Section 7.5. We report conclusions in Section 7.6.

Task	Dataset	Concept Instances	Concept Type	Concepts /Query	Images /Query	Split	Queries	Pool	Concepts	Contexts	Contexts /Concept	Concepts /Context
pers TIR **pers TIG**	ConCon-Chi	**yes**	**chimeric** & common	**≤ 3**	**6.07**	val	42	165	3	20	14.34	2.15
						test-unseen	**986**	**4008**	**17**	**101**	**50.53**	**8.50**
						test	**1084**	**4008**	**20**	**101**	**46.65**	**9.24**
pers TIR	PerVL DF2	**yes**	fashion	1	1	val	229	229	**50**	**229**	5.58	1.00
						test	221	221	**50**	**221**	4.42	1.00
pers TIG	DreamBooth	**yes**	common	1	0	test	**750**	0	**30**	35	25	**21.43**
	CC101	**yes**	common	**≤ 2**	0	test	**3232**	0	**101**	**597**	**36.36**	**6.15**
ZS-CIR	FashionIQ	no	fashion	1	1	val (avg)	**2005**	**5179**	**1442**	**1994**	1.39	1.00
						test (avg)	**2039**	**5179**	**1454**	**2030**	1.40	1.00
	CIRR	no	open	1	1	val	**4184**	2297	**2165**	**4157**	1.93	1.00
						test	**4148**	2315	**2178**	**4135**	1.90	1.00
	CIRCO	no	open	1	**4.53**	val	**220**	**123403**	**220**	**220**	1	1
						test	**800**	**123403**	**798**	**796**	1.00	1.00

Table 7.1 Comparison of related datasets. PerVL DF2: PerVL DeepFashion2; CC101: CustomConcepts101. For FashionIQ we report the average number of images per split (shirt, dress, toptee). Dataset aspects that fulfill the criteria discussed in Section 7.3 are in **Bold**. ZS-CIR datasets reported to show the setting difference with personalized TIR/TIG.

7.2 Related Work

Learning Out Of Vocabulary (OOV) words. Learning textual concept representations from images can be seen as the multi-modal version of the NLP task of learning OOV word embeddings from a few examples (Bahdanau et al., 2017) or the word definition (Herbelot and Baroni, 2017; Hill et al., 2016; Ruzzetti et al., 2022). We took inspiration from the Chimera dataset (Lazaridou et al., 2017), which differentiates the problem of learning to associate new words with existing meanings (named entities, synonyms, aliases) from the one of learning new meanings, and models the latter with chimeric words that incorporate two unrelated concepts in a single one.

Personalized Text-to-Image Retrieval (TIR). Cohen et al. (2022) introduced "Personalized Vision & Language" (PerVL), a setting where the vocabulary of a VL model is expanded with pseudo-tokens whose embeddings are learned from few images of user-specific concepts. The benchmark includes a retrieval and an instance segmentation dataset obtained by re-annotating images respectively from DeepFashion2 (Ge et al., 2019) and YouTube-VOS (Xu et al., 2018) with concept-context captions. PerVL includes many concept instances, which are however restricted to fashion or common items; moreover, it counts very few contexts per concept and a single concept per context, suffering from context bias. To tackle the proposed benchmark the authors propose PALAVRA, which we evaluate in this study.

The setting of personalized retrieval has been extended to videos (Korbar and Zisserman, 2022; Yeh et al., 2023). Parisot et al. (2023) proposed a similar approach to transfer CLIP (Radford et al., 2021) to a downstream image classification task by optimizing the class names.

Zero-Shot Composed Image Retrieval (ZS-CIR). This task can be seen as one-shot personalized TIR, where the reference image is the concept, and the relative caption the context. Thus personalization methods as PALAVRA have been evaluated for ZS-CIR (Baldrati et al., 2023). Similarly, we consider methods as Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023) for evaluation on the proposed personalized TIR benchmark. However, we remark that the two settings are different. In recent ZS-CIR datasets (e.g., CIRR (Liu et al., 2021b) and CIRCO (Baldrati et al., 2023)), images are drawn from an open domain (respectively NLVR^2 (Suhr et al., 2019) and COCO (Lin et al., 2014)), thus concepts are defined at a semantic category level. Moreover, the reference image typically contains multiple elements. This aspect poses the additional challenge of understanding to which image element the relative caption refers to (see, e.g., (Chen et al., 2023)), and whether an image actually represents the same concept or not. Differently, in the considered personalization setting, concepts are typically instances, clearly identifiable in the provided image examples. FashionIQ (Wu et al., 2021) is another CIR dataset focusing on kinds of fashion items.

Personalized Text-to-Image Generation (TIG). This task was proposed in Gal et al. (2023a) and is the generative counterpart of personalized TIR. The authors presented Textual Inversion (Gal et al., 2023a), a method that expands the vocabulary of a frozen text-to-image model (Latent Diffusion (Rombach et al., 2022)) with user-specific concept embeddings learned from few images. In Ruiz et al. (2023) the authors present DreamBooth, a method that selects a rarely-used token and binds it to the concept by fine-tuning the text-to-image model on the concept images. Subsequent methods propose improvements over these two (Gal et al., 2023b; Kumari et al., 2023; Tewel et al., 2023). The datasets introduced by these works (see, e.g., DreamBooth and CustomConcepts101 (Kumari et al., 2023; Ruiz et al., 2023)), are constituted by concept training images and a list of prompts for evaluation, but do not contain any real image representing such prompts. The validation is thus carried out by comparing a generated image with the training images (to measure concept fidelity) and with the context in the prompt (to measure context fidelity). Differently, typical validation metrics for generative models measure a distance between the population of real and generated samples (see, e.g., (Han et al., 2023; Kynkäänniemi et al., 2019; Naeem et al., 2020). Since in ConCon-Chi a set of real image realizations is provided for each prompt, we show how the application of these distance measures can improve the validation of personalized TIG methods.

7.3 Concept-Context Chimera Benchmark

We present ConCon-Chi and compare with existing datasets in Section 7.3.1, then describe its acquisition process in Section 7.3.2.

7.3.1 Dataset overview

Since personalized TIR/TIG are few-shot tasks, we compare in Table 7.1 with related benchmarks in terms of validation/test splits.

ConCon-Chi test split includes 1084 queries and a pool of 4008 images which are treated as ground-truth images for TIR and image realizations for TIG ($\sim$6 per query). Each query was generated by composing up to 3 concepts and a context from a set of 20 concepts and 101 contexts. The average number of contexts associated to each concept is indicated under CONTEXTS/CONCEPT and gives an idea of the variety of situations in which a concept is required to be retrieved or generated.

In Table 7.1 we also highlight the difference between personalization and ZS-CIR datasets. These latter do not explicitly deal with concepts and contexts, but each reference image and relative caption that compose a query are counted as a different concept and context. Thus their number of concepts and context is higher than in personalized tasks and the ratios CONCEPTS/CONTEXT and CONTEXTS/CONCEPT are close to 1. Differently, to evaluate the capability of personalization methods to combine the learned concepts with contexts, CONTEXTS/CONCEPT should be high and, to avoid context bias, CONCEPTS/CONTEXT should be high as well.

The most similar dataset to ours is CustomConcepts101, which however does not contain real images for evaluation (see POOL and IMAGES/QUERY in Table 7.1).

Concepts. In Figure 7.2 we show 10 of the 20 concepts in the dataset (the complete set is in supp. material Figure 7.12). As in Ruiz et al. (2023); Tewel et al. (2023), we include animate and inanimate entities (puppets and tools, accessories, clothes) with the aim of creating rich interactions. There are 6 chimeric concepts, which are animal or alien puppets crafted out of tools or accessories, and 14 common concepts. Among these latter, we include instances of the same categories composing the chimeras, which are thus visually and semantically similar. These are hard negatives especially when appearing in the same context (e.g., BIRDYER and MYSPRAYER in Figure 7.1) and enhance the compositionality assessment (Ma et al., 2023).

Concept names are invented and provided for completeness since are not used by the considered methods. Differently, in our evaluation (e.g., Table 7.2) we consider feeding

Figure 7.2 **Example concepts in ConCon-Chi[1].** Four chimeric and six common concepts (hard negatives outlined with same color).

a pre-trained VL model with a description of the concept as zero-shot baseline, thus we provide the adopted descriptions for reproducibility since words choice was empirical (supp. material Table 7.5). *Discriminative* descriptions were formulated to be a competitive baseline, by thinking to a minimal sentence discriminating the concept from others in the concept set. *Coarse* descriptions are category-level and model the typical case where the concepts to be discriminated from are unknown (e.g., "bag", but there are two bags in the concept set). *Rich* descriptions aim to verify whether enriching the *Discriminative* with visual details degrades performance; specifically, they extend the *Discriminative* descriptions with additional details such as color, shape and parts (e.g., "snake box-headed puppet" $\rightarrow$ "green box-headed snake puppet with round eyes and pink tongue").

Contexts. Contexts are grouped into 9 kinds (one example per kind in Figure 7.1). Inspired by Ruiz et al. (2023), these are structured in concept modifications (accessorizations, property modifications, renditions) and relationships (actions, interactions, re-contextualizations). We describe them and present their distribution in supp. material Figure 7.13a.

Similarly to Ruiz et al. (2023); Tewel et al. (2023) we include general and specific contexts: general contexts are applicable to all or most concepts (e.g. re-contextualization in Figure 7.1) and strongly contribute to avoiding context bias; specific contexts are typical of a concept category and aim to increase the difficulty of distinguishing between hard

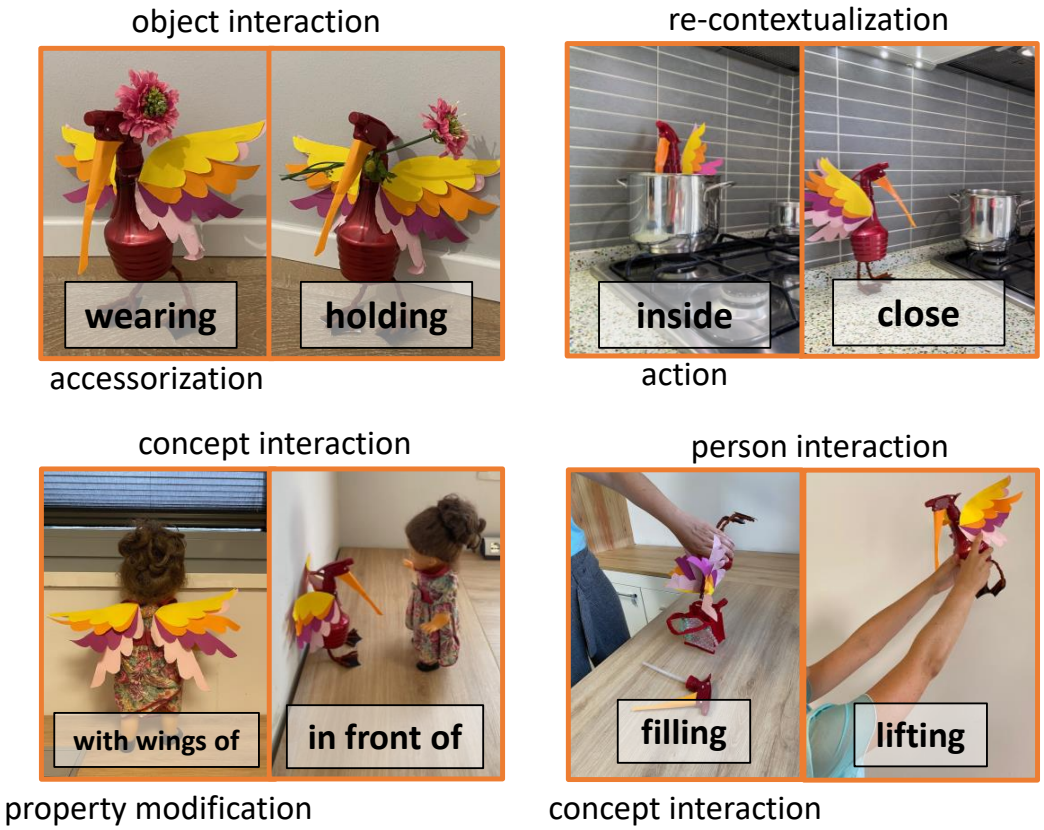


Figure 7.3 **Fine-grained contexts.** Examples queries for the concept BIRDYER, in which the recognition of the co-occurrence of elements does not suffice for retrieval.

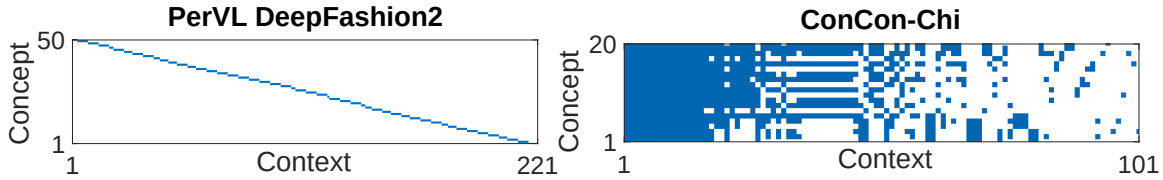


Figure 7.4 **Concept-context matrices.** Blue cells mark the concept-context combination appearing in the dataset (*test* split).

negatives (e.g. the action *containing* for bags-like objects). When a context specifies an interaction between a concept and an entity (object, person, another concept), a retrieval method could perform well simply by detecting their co-occurrence. To avoid this we included contexts specifying different interactions with the same entities (see Figure 7.3). A particular interaction is the one between two learned concepts. Concept-concept interactions have been studied in personalized TIG (Kumar et al., 2022; Tewel et al., 2023). In Section 7.4 we evaluate this aspect in personalized TIR.

Concept-context structure. In Figure 7.4 we compare the concept-context structure of our dataset with PerVL DF2. A matrix cell is Blue when a query formed by the corresponding context and concept exists in the dataset. While our dataset contains approximately half of the concepts and contexts, our concept-context matrix is denser, reflecting a higher number of queries (1084 vs. 221, see Table 7.1). Specifically, in PerVL DF2 each context is coupled with a single concept, while in our dataset this only occurs for 8 contexts, with 16 contexts combining with every concept. Moreover, in ConCon-Chi each concept is composed with 46.64 contexts on average vs. 4.42 in PerVL DF2 (few cells per row): similarly to context bias, concept bias limits the compositionality assessment since it makes it difficult to determine to what extent the retrieval method understands the context.

7.3.2 Acquisition and Annotation

Acquisition. First we gathered the objects and crafted the chimeric concepts; then we moved to photograph each concept in the designed contexts. We arranged the scene to represent the query and shot multiple pictures by varying the viewpoint. Any private and sensitive information was removed from the scene; images containing humans were cropped to remove the face and make the subject unidentifiable. We provide the distribution of environments in supp. material Figure 7.13b and details about the creation of renditions in supp. material Section 7.7.3.

Annotation. Annotations consist of query-image associations. A first step consisted of labelling each image with the represented concepts and context. A second step was carried out to remove false negatives in the annotations. These happen whenever the ground-truth (GT) images of a query that is more specific, are not included among the GT of a query that is more generic (query overlap). A similar problem happens when GT images of a query contain

[1]Any representation of trademarks, trade names, logos, domain names such as any other distinguishing marks appearing in this dataset is purely random, and it is used exclusively for scientific and non-commercial purposes; therefore, the relevant representation cannot be understood as an expression of an opinion or an indication or a precondition for taking decisions.

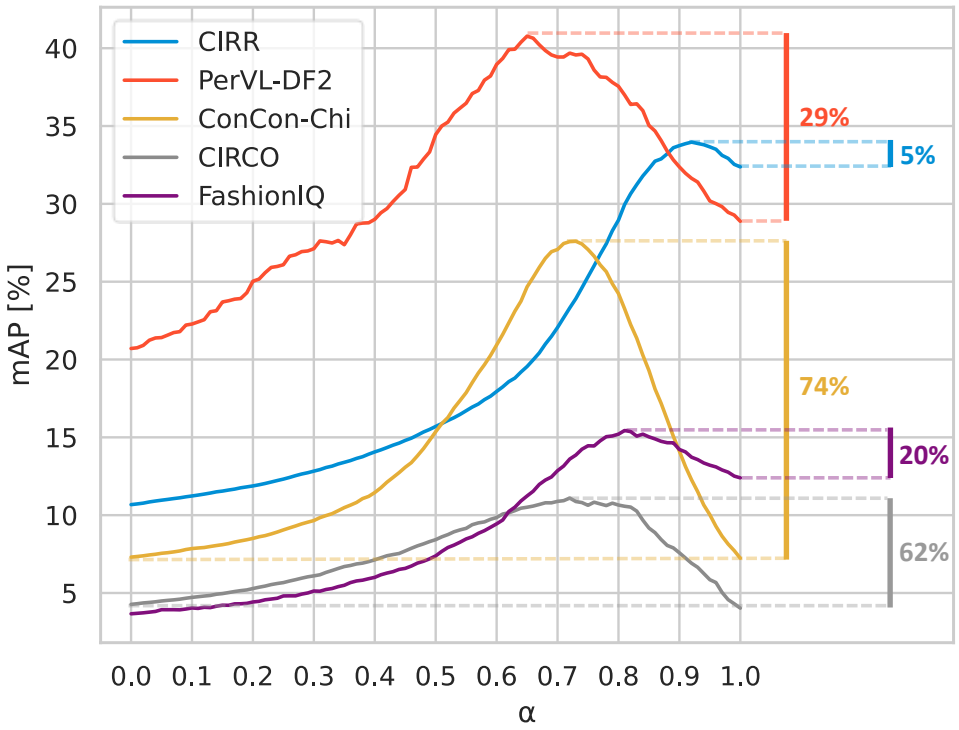


Figure 7.5 **Concept and context biases.** Retrieval performance by varying the relative weight (α) of the concept and the context in the query. Lateral bars: performance drop when using either only the concept (α=0) or the context (α=1), relative to peak performance.

content that also corresponds to other queries (image overlap). In benchmarks created by re-annotating existing datasets of images "in the wild" it is not possible to control overlaps and an exhaustive check is unfeasible. To this end, CIRR (Liu et al., 2021b) and CIRCO (Baldrati et al., 2023) designed approximated procedures to ensure that false negatives are respectively absent in sub-pools of images or below an estimated percentage. We accounted for the problem since the dataset design by controlling and minimizing overlaps. The GT images were then assigned following concept and context overlap and manually checking those of queries where we were aware of possible image overlap. As a result, to the best of our knowledge, the set of 1084 queries is free from false negatives with respect to the pool of 4008 images. We report the number of GT images per query in supp. material Figure 7.14.

7.4 Personalized Text-to-Image Retrieval

We define the TIR benchmark task on ConCon-Chi and compare it with related benchmarks in Section 7.4.1. We describe the considered methods in Section 7.4.2. We present and analyse results respectively in Section 7.4.3 and Section 7.4.4.

7.4.1 Benchmark Task

Train and test splits. Each concept in the set of 20 is trained independently on 1 to 5 images where the concept is standing in front of some background (*train* split, examples in Figure 7.2). At test time, for each query in the *test* split we rank the images in the pool according to their similarity with the query. Training backgrounds do not appear in the *test* split. Since the concepts are not trained jointly, similar concepts play the role of hard negatives. In supp. material Section 7.7.4 we introduce other splits not used in this paper.

Metrics. For each query we evaluate the rank of the first GT image (mean Reciprocal Rank, mRR) and whether this is among the top-k (recall rate, R@k); then we evaluate the rank of all GT images (mean Average Precision (Beitzel et al., 2009), mAP) and of GT images up to the top-k, mAP@k (Baldrati et al., 2023). See supp. material Section 7.8.1 for definitions.

Concept and context biases. We first aim to compare the presented benchmark with existing ones in terms of context and concept biases. To quantify the importance of attending the concept and the context for correct retrieval, we adopt the experimental setup proposed in Saito et al. (2023) and model the concept-context interaction by computing a query embedding q as weighted sum of c, the CLIP (Radford et al., 2021) embedding of the context (output of CLIP text encoder, ViT-L14 backbone), and i, the average CLIP embedding of the concept training images (output of the vision encoder): $q = (1-\alpha)\cdot i + \alpha \cdot c$ with $\alpha \in [0,1]$. In Figure 7.5 we compare the retrieval performance by varying α on ConCon-Chi, PerVL-DF2, CIRCO, CIRR and FashionIQ. For PerVL DF2 and ConCon-Chi we consider the available training images and for each dataset we use the *test* split if available, the *validation* otherwise (see Section 7.8.2 and Table 7.1 for details). When in ConCon-Chi multiple concepts appear in the query we average their embeddings.

For three datasets the performance achieved by just putting the context in the query (α=1) is higher than 70% of the peak performance, indicating that the probability of retrieving the correct context-concept combination is relatively high even when ignoring the concept (context bias). Differently, in ConCon-Chi we observe low performance for α=1, and a drop of 74% from the peak performance with α=0 (only the concept in the query). Only CIRCO

	Method	mAP [%]	mRR [%]	R@1 [%]
k=0	*Coarse*	16.83	24.21	14.48
	Discriminative	*30.16*	*43.16*	*31.92*
	Rich	*27.65*	*40.58*	*29.98*
k=1	PALAVRA	22.56 ± 1.29	34.39 ± 1.68	24.59 ± 1.94
	Pic2Word	25.23 ± 1.20	37.16 ± 1.76	26.35 ± 1.85
	SEARLE	28.16 ± 0.55	41.07 ± 0.92	31.16 ± 0.94
k=5	PALAVRA	23.59	35.99	26.75
	Pic2Word	26.39	38.62	27.68
	SEARLE	**30.74**	**43.83**	**33.49**

Table 7.2 **Personalized TIR benchmark.** Performance of CLIP baselines (k=0) and personalized TIR (PALAVRA) and ZS-CIR methods (Pic2Word, SEARLE) on the retrieval task in ConCon-Chi. The best method per metric is in **Bold**; *Discriminative* and *Rich* baselines (oracles) are highlighted with *asterisks*.

exhibits a similar trend, with lower peak performance possibly because of the intrinsic difficulty of the proposed ZS-CIR task and the large image pool.

7.4.2 Methods

We compare the personalized TIR method PALAVRA (Cohen et al., 2022) with two ZS-CIR methods, Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023), on the defined TIR benchmark. Since, differently from ZS-CIR, in our setting multiple concept example images are available, for the two latter methods we average the generated token embeddings to create the concept embedding. All methods rely on CLIP ViT-L14 backbone with same input pre-processing. The methods learn a textual token embedding that expands CLIP vocabulary and assign it an arbitrary textual identifier such that at inference they replace the learned embedding whenever this is encountered in an input query. We used the code released by the authors (see Section 7.4.2 for details).

For each method we provide a brief description of how it works followed by some details on the choices we took to ensure a fair comparison.

PALAVRA (Cohen et al., 2022). This method uses a Deep Set function (Zaheer et al., 2017) to map the 5 CLIP image embeddings of the concept to a token embedding (inverse mapping), as illustrated in Figure 7.6. The function is pre-trained on thousands of frequent concepts from COCO (Lin et al., 2014) and used to provide an initial token value for the concept at hand. This is then fine-tuned with a so-called cycle contrastive loss, such that the CLIP text embedding of a template sentence containing the token is pushed closer to the average CLIP image embedding of the 5 images (cycle term) and farther from the embedding of a same sentence with the concept type replacing the token (contrastive term).

We used the code released by the authors [2] and retrained the inverse mapping following author instructions. When doing so, we replaced the ViT-B/32 CLIP backbone with the larger ViT-L/14 for fair comparison with other methods. When fine-tuning the token embedding, we adopted hyper-parameters suggested by the authors and when using a single image example, we fed the Deep Set function with 5 copies of the single image embedding. As concept types in the fine-tuning, we used the *Coarse* descriptions.

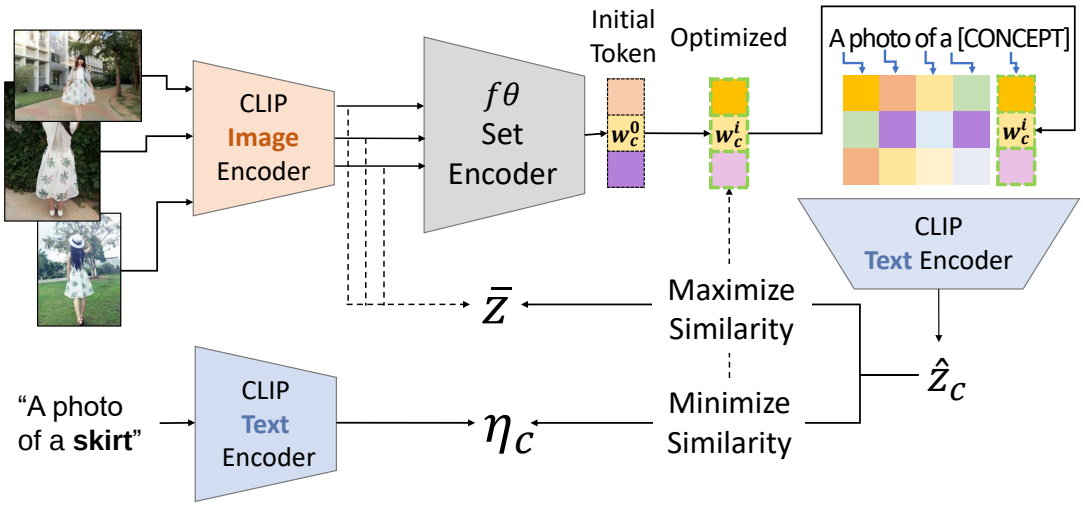


Figure 7.6 **PALAVRA architecture.** Architectural schematic of the personalized TIR baseline PALAVRA.

Pic2Word (Saito et al., 2023) and SEARLE (Baldrati et al., 2023). These are two ZS-CIR methods which, as PALAVRA, learn a inverse mapping function for CLIP. In this case this is a three-layer MLP that maps a CLIP image embedding (input) to a corresponding token embedding (output) and is learned in a self-supervised fashion.

Pic2Word trains the function on 3M images from CC3M(Sharma et al., 2018) with a cycle contrastive loss that uses a given CLIP image embedding as cycle term and others as contrastive term. In SEARLE, first, a token embedding for each of 100K images in the unlabeled test split of ImageNet1K (Russakovsky et al., 2015) is optimized similarly to PALAVRA's fine-tuning, with a regularization term such that the embedding is also kept close to the textual embeddings of concepts represented in the image. Then, since using this optimization procedure at evaluation time would be computationally expensive, SEARLE learns a mapping network that produces similar embeddings to the ones learnt through back-propagation. This mapping network is trained with a contrastive loss over the 100K

[2] https://github.com/NVlabs/PALAVRA

optimized tokens, regularized in the same way. At inference time SEARLE architecture is basically identical to Pic2Word. The SEARLE method is shown in Figure 7.7.

We use both methods by downloading pre-trained weights of the mapping functions based on CLIP ViT-L14, officially provided by the authors (Pic2Word[3], SEARLE[4]). Differently from ZS-CIR, in our setting 5 concept images are available, thus we average the generated token embeddings to create the concept embedding.

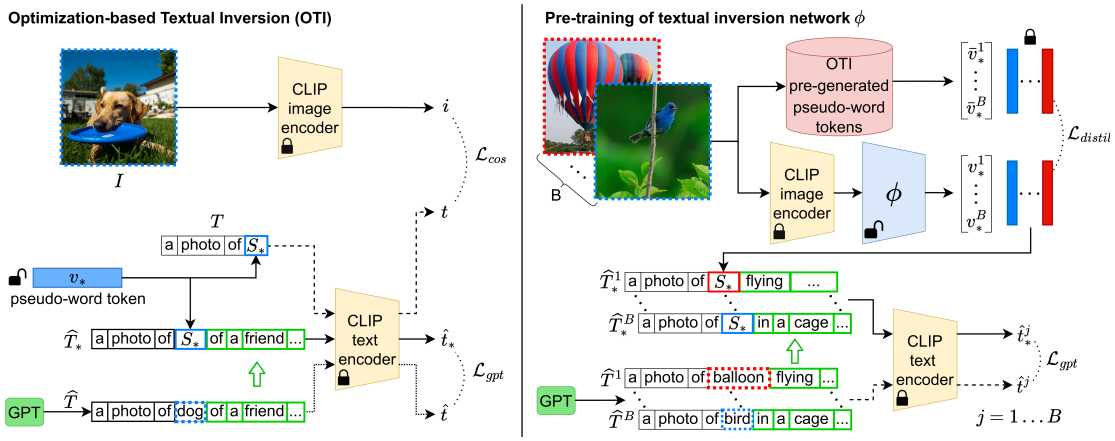


Figure 7.7 **SEARLE architecture.** Architectural schematic of the SEARLE method.

7.4.3 Benchmark Results

Results. In Table 7.2 we evaluate each method with k=1 and k=5 training images per concept (for k=1 we report mean and standard deviation over the 5 images). We also report, as baselines that do not use any image (k=0), the performance achieved by feeding CLIP with queries where the concept identifier is replaced with the *Coarse*, *Discriminative* or *Rich* descriptions introduced in Section 7.3.1.

As expected, *Coarse* descriptions provide a lower bound, since they are shared among the concepts of same category to simulate descriptions that are not tailored to the discriminative task at hand. Differently, *Discriminative* and *Rich* descriptions provide a competitive baseline (higher than PALAVRA and Pic2Word). They represent an oracle, since were formulated by discriminating every concept from the others, thus accessing information which is unavailable to other methods that learn each concept independently. In this respect, we note that adding visual details degrades performance. Interestingly, SEARLE outperforms the *Discriminative* baseline thus being the best method, also because it exhibits a smaller standard deviation in

[3]https://github.com/google-research/composed_image_retrieval
[4]https://github.com/miccunifi/SEARLE

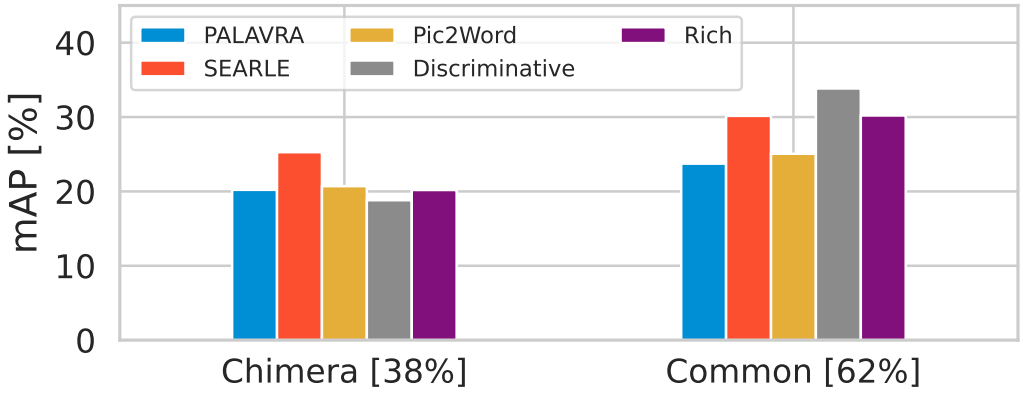


Figure 7.8 **New-meaning learning.** Retrieval performance on queries containing at least one chimeric concept vs. common concepts (percentage of each query kind in the dataset in brackets).

Method	All Concepts	Chimeric Concepts	Common Concepts	Contexts
Coarse	29.38	09.51	37.89	42.11
Discriminative	*66.48*	*60.61*	*68.99*	40.92
Rich	*84.54*	*85.54*	*84.11*	29.81
PALAVRA	**91.52**	**91.15**	**91.67**	28.56
Pic2Word	50.04	51.32	49.49	**50.12**
SEARLE	76.58	82.25	74.15	40.47

Table 7.3 **Concept-context compositionality.** Performance (F1 score [%]) of recognition of concepts and contexts in the retrieval task of Table 7.2.

the 1-shot scenario and a larger gain when more images are provided. This confirms that SEARLE outperforms the other two methods not only in ZS-CIR (Baldrati et al., 2023) but also in the personalized TIR setting. We report more metrics in supp. material Table 7.7.

7.4.4 Analysis of Results

We analyse the performance achieved in the benchmark in terms of new-meaning learning and compositionality. For this study we consider the methods trained on 5 images.

New-meaning learning. In Figure 7.8 we report the mAP of Table 7.2 separately on queries containing at least one chimeric concept and common ones. We see that all methods and baselines achieve a lower performance when retrieving queries containing a chimeric concept. We then inspected whether this can be explained with a misclassification of the concept or the context, or their combination. For each query therefore we considered the first N ranked images (N equal to the number of GT images) and marked the concept in the query as ground-truth and the concept in each retrieved image as a prediction. We computed

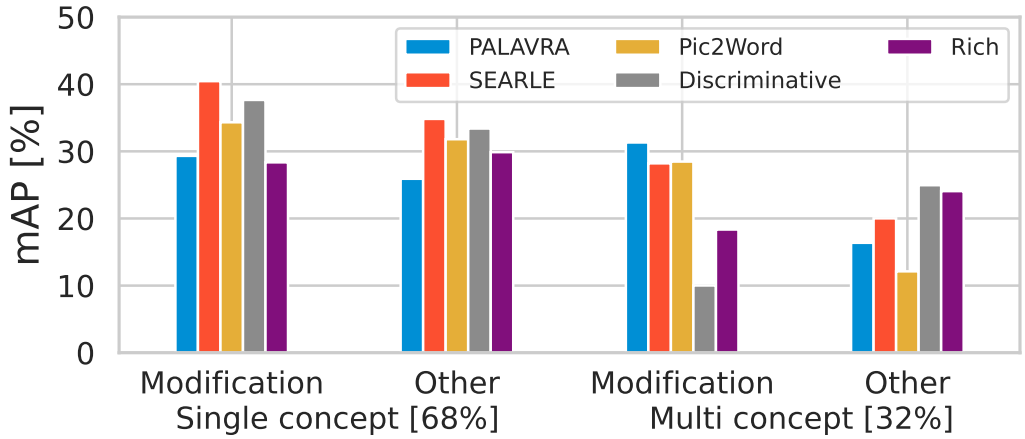


Figure 7.9 **Concept-concept compositionality.** Performance on single- and multi-concept queries, comparing "accessorization" and "property modification" ("modification") kinds versus the rest of the kinds ("other").

the F1 score for each concept by accumulating the predictions from the queries containing that concept and we report in Table 7.3 the average scores over all, chimeric and common concepts. Please note that we restricted this analysis to single-concept queries and discarded ranked images containing multiple concepts.

In Table 7.3 we observe that personalization methods achieve equal or better classification performance for chimeric with respect to common concepts. We then conclude that, while these methods learn to recognize the visual appearance of the chimeras, they struggle more at retrieving these new meanings in composition with known contexts.

Concept-context compositionality. As for the concepts, we computed the classification performance in a similar way for the contexts, and report it in Table 7.3. We see that PALAVRA achieves the best concept and the lowest context classification performance. We deduce that its low retrieval performance in the benchmark must be due to the poor compositionality properties of the learned tokens. Conversely, Pic2Word excels at recognizing the contexts, but cannot learn discriminative tokens. Thus, the good retrieval performance of the *Discriminative* baseline and of SEARLE in Table 7.2 seems explained by their capability to trade-off and combine context and concept recognition. This seems confirmed by the fact that the *Rich* descriptions improve concept recognition at the expense of the context, probably since longer descriptions outweight the rest of the query. In the case of SEARLE the ability to generate meaningful and compositional embeddings can be attributed to its regularized training. In fact, the embeddings produced by the mapping function are forced to reside on the CLIP token embedding manifold, making them meaningful when combined

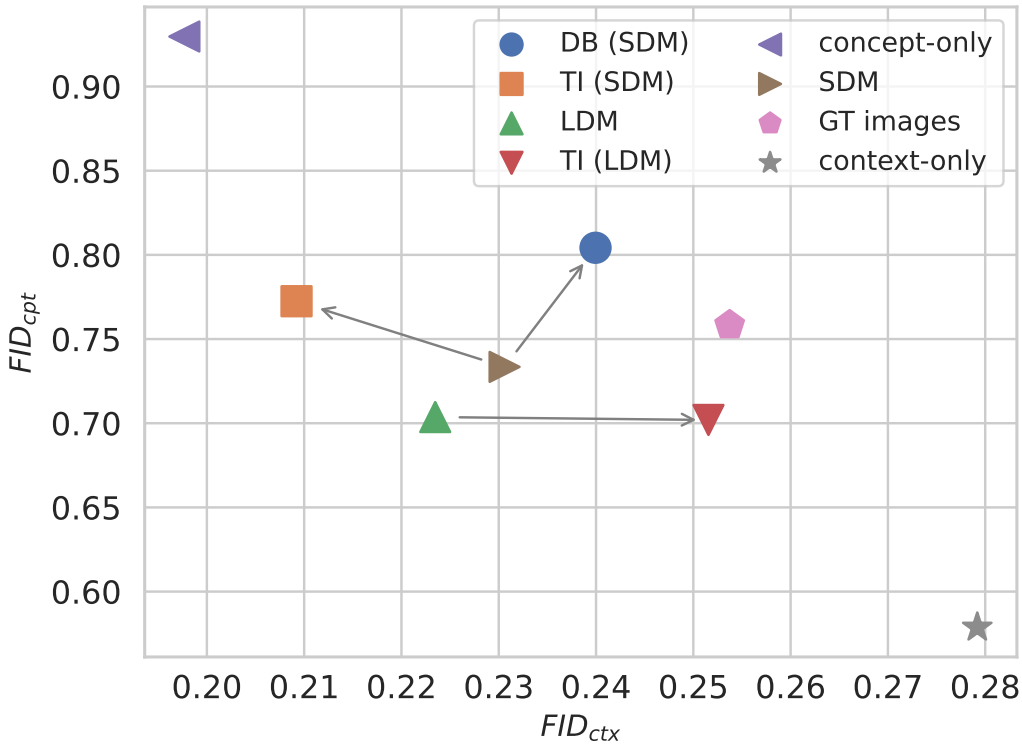


Figure 7.10 **Personalized TIG benchmark.** Scatter plot of the FID_{cpt}/FID_{ctx} trade-off for the considered methods, baselines and upper bounds on the generation task ConCon-Chi.

with other vocabulary tokens. By contrast, Pic2Word and PALAVRA do not enforce the learned token to lie on the textual manifold which can explain a loss of compositionality between the concept and context tokens.

Concept-concept compositionality. We finally investigated whether performance also depends on the context kind. In Figure 7.9 we report the mAP of Table 7.2, separately on queries containing a single and multiple concepts and, in each group, containing a concept modification ("property modification" or "accessorization") or another kind of context. As expected, multi-concept queries are more challenging. On these we observe an interesting trend: description-based baselines drop performance on concept modifications, while the personalization methods drop performance on the other kinds. Multi-concept queries with concept modifications are mostly the ones where a part of a concept is worn or applied to another one (an example in Figure 7.3 where the wings of BIRDYER are applied to MYDOLL). These queries require knowing visual information on the concept parts, thus explaining why the baselines fail in these cases. Conversely, the other kinds of multi-concept queries typically require the detection of the co-occurrence of the concepts and their relationship (e.g., in Figure 7.3, "MYDOLL in front of BIRDYER"). To achieve this, the concept textual

		k=3		k=10			
		Density ↑	Coverage ↑	Density ↑	Coverage ↑	FID_{cpt}	FID_{ctx}
SDM	Common	2.66	0.12	8.84	1.33	0.74	0.23
	Chimeric	0.78	0.02	2.40	0.21	0.73	0.23
TI	Common	3.43	0.23	11.56	**2.93**	0.71	**0.25**
	Chimeric	2.71	0.15	5.43	0.90	0.69	**0.25**
DB	Common	**6.55**	**0.28**	**18.98**	2.65	**0.80**	0.24
	Chimeric	4.81	0.14	11.49	1.09	0.79	**0.25**

Table 7.4 **Comparing generated and GT images.** We consider TI and DB with SDM. The best for each metric is in **Bold**. Coverage and Density are in [%] (Density not upper bounded by 100).

representations must exhibit robust compositionality properties. Thus, the tokens learned by the methods do not seem to retain the same compositionality properties of the tokens in the original vocabulary (which form the descriptions).

7.5 Personalized Text-to-Image Generation

We define the TIG benchmark task in Section 7.5.1. We describe the considered methods in Section 7.5.2, present results in Section 7.5.3, and analyse them in more detail in Section 7.5.4.

7.5.1 Benchmark Task

Train and test splits. We train each concept on the same 5 images used for TIR and at test time we generate 4 images per prompt. We restrict the evaluation to the single-concept prompts in the *test* split (which are 735).

Metrics. We report the two metrics introduced in Gal et al. (2023a) and adopted by following works. These rely on CLIP and compute, for each prompt, the average pairwise cosine similarity between the visual embeddings of the generated images and of, respectively, the visual embeddings of the training images (fidelity-concept or FID_{cpt} as in (Ruiz et al., 2023)) and the textual embedding of the context in the prompt (fidelity-context or FID_{ctx}). Personalized TIG methods are expected to trade-off the two metrics by learning to represent the concept appearance while retaining the capability to represent contexts.

7.5.2 Methods

As in personalized TIR, the considered methods assign a textual identifier to the concept and when this is encountered in the prompt they load a corresponding learned token embedding (Textual Inversion, TI (Gal et al., 2023a)) or model (DreamBooth, DB (Ruiz et al., 2023)). For DB we adopted the Diffusers library (von Platen et al., 2022) with a Stable Diffusion Model (SDM) (Rombach et al., 2022) as pre-trained text-to-image model (called DB(SDM)). For TI we adopted the code by the authors with either a Latent Diffusion Model (LDM) (Rombach et al., 2022) or, by following Tewel et al. (2023), the same SDM as for DB (called TI(LDM) and TI(SDM)). Details in Section 7.5.2.

DreamBooth (DB) (Ruiz et al., 2023). This method fine-tunes a pre-trained TIG model on the example images in order to bind a textual identifier (a chosen rare token followed by the concept class) to the concept appearance. A class-specific prior preservation loss acts as regularizer by forcing the model also to keep generating images of instances of the concept class when the input sentence does not contain the rare token but just the class.

We adopted the implementation provided[5] in the Diffusers library (von Platen et al., 2022), release 0.18.0, with the Stable Diffusion Model (SDM) checkpoint v1.4[6]. We then fine-tuned the text encoder alongside the UNet, using the *Coarse* descriptions at initialization, with a learning rate of 1e-6.

Textual-Inversion (TI) (Gal et al., 2023a). This method applies an approach similar to PALAVRA and optimizes a token embedding in the vocabulary space of a frozen TIG model (by feeding it with template sentences containing the token and asking it to generate images like the 5 examples). The authors show that a single token suffices to generate a concept faithfully and highlight the advantage of the approach to retain the knowledge of the pre-trained model.

We adopted the officially released code[7], using the 1-token *Coarse* descriptions for the token initialization. We either adopted the Latent Diffusion Model (LDM) (Rombach et al., 2022)[8] by following authors' instructions, or, by following more recent work (Tewel et al., 2023), the same SDM used for DB, with the parameters that the authors report for LDM (learning rate of 5e-3) and by applying the same procedure as for DB to select training steps.

Number of steps in the concept optimization. Since we observed performance variations depending on the number of steps, we devised an automatic procedure to choose

[5]https://github.com/huggingface/diffusers/tree/main/examples/dreambooth
[6]https://huggingface.co/CompVis/stable-diffusion-v-1-4-original
[7]https://github.com/rinongal/textual_inversion
[8]https://github.com/CompVis/latent-diffusion

the optimal number and applied it to both DB and TI. We created a minimal validation set composed of 4 prompts ('a photo of * on a beach', 'a photo of * on the moon', 'a photo of * with a cat', 'a photo of a yellow *'). We avoided using the larger proposed *validation* set since, while we defined it for completeness and possible future uses, we preferred preserving the 5-shot nature of the considered personalization setting.

The score to maximize was chosen empirically among different versions, by comparing the best checkpoints selected automatically, with checkpoints selected manually (by qualitatively comparing the generated images at each validation iteration), for each of the 20 concepts. The best score S resulted being S=min-max(FID_{ctx}), subject to: min-max(FID_{ctx}) < min-max(FID_{cpt}). Namely, we first scale each metric such that it is in $[0, 1]$ along the optimization (min-max). Then, we maximise the capability to generate an image of the context (we maximize FID_{ctx}), subject to the fact that the generated images are also close to the concept image examples (subject to FID_{ctx}) < FID_{cpt}). This provided slightly better results than the harmonic mean of the FID_{cpt} and a normalized FID_{ctx}, adopted in (Tewel et al., 2023).

While the ground-truth step annotation was possible for DB, we could not easily identify a learning trend in TI. Thus, we validated the procedure on the former method and then applied it also to the latter.

7.5.3 Benchmark Results

Results. In Figure 7.10 we show the fidelity-concept/context trade-off (numbers in supp. material Table 7.8). To quantify these metrics, as in Gal et al. (2023a) we compare with upper bounds. For the fidelity-concept, the upper bound returns always the 5 training images irrespective of the prompt (concept-only). For the fidelity-context, the upper bound are images generated by the pre-trained SDM when fed as input only with the context of the prompt (context-only). We also evaluate the baseline performance of the LDM or SDM pre-trained models by replacing the concept identifier in the prompt with its *Rich* description. As reference, we report also the metrics for the GT images available in ConCon-Chi.

We first observe that TI(LDM) does not improve the fidelity-concept over LDM, but improves the fidelity-context. This can be explained by considering that the *Rich* description in LDM tends to outweight the context. Conversely, TI(SDM) improves the fidelity-concept of SDM, while degrading the fidelity-context. This behaviour has also been observed in recent work (Tewel et al., 2023) and may be due to the learned token over-fitting the context of the training images. A better learning behaviour is provided by DB(SDM), which

improves both metrics over SDM. Notably, compared with the GT images, DB shows a higher fidelity-concept and lower fidelity-context.

7.5.4 Analysis of Results

We leverage the availability of GT images to inspect the performance of DB, TI and their pre-trained model SDM with two metrics proposed for evaluation of text-to-image generation (Naeem et al., 2020): Density as measure of realism (fidelity) and Coverage as measure of diversity (how well generated images span the real images manifold). We created one real and one generated manifold per concept as the union of all prompts for that concept (similarly to Han et al. (2023) Appendix L). In Table 7.4 we report the metrics for two values of the parameter k, averaged over chimeric and common concepts. For all methods, Density and Coverage are higher on common concepts than on chimeric ones. This is in line with what has been observed in Ruiz et al. (2023); Tewel et al. (2023) that more common objects are easier to generate since these methods leverage the pre-trained knowledge about the concept category. Interestingly, such difference is not evident from the $\text{FID}_{\text{cpt/ctx}}$ metrics typically adopted in personalised TIG. We observe that DB has best Density, while in terms of Coverage there is a smaller gap with TI. Thus DB achieves relatively better realism, but diversity remains low. We conclude that leveraging the availability of a real population of images allows for a more thorough evaluation of personalised TIG: there seems to be large room for improvement, with new-meaning concepts posing more challenges.

Qualitative examples. We report some cherry-picked examples in Figure 7.11. We observe that DB represents the concepts more accurately and combines them more nicely with the context than TI, which sometimes forgets the concept (First Row) and sometimes the context (Third Row). It can also be noticed how common concepts are learned very accurately by DB (First and Third Row), while chimeric ones are not. As expected, using a visually *Rich* description (Third Column) is not enough for personalized generation, especially on chimeric concepts (Second and Fourth Row), which lack more distinctive features than the common ones.

7.6 Conclusion

We present a new dataset called ConCon-Chi for the evaluation of personalized TIR and TIG. The dataset models novel concepts as chimeric objects and by adopting a concept-context matrix structure allows to study the learning of new meanings in terms of their

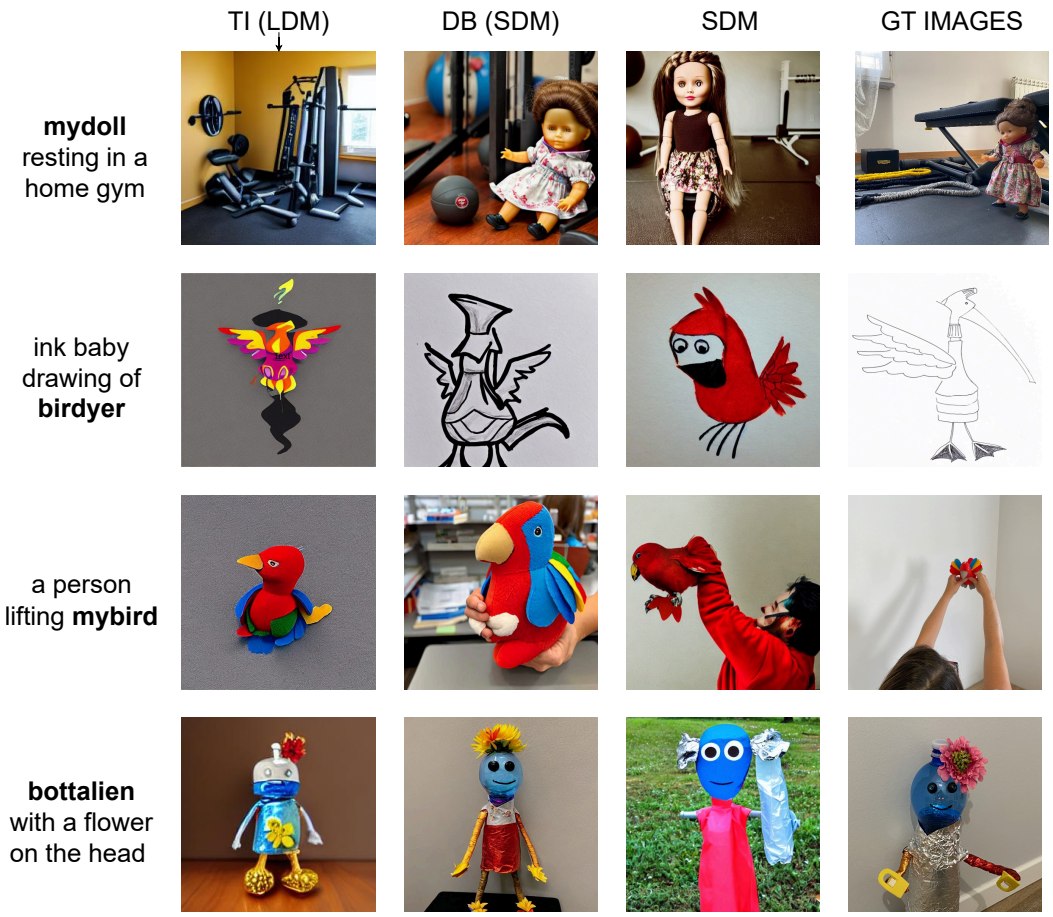


Figure 7.11 **Qualitative results for personalized TIG.**

compositionality properties with known ones. Our analysis on retrieval showed that current methods struggle at composing these new concepts with known contexts, and also together. A similar issue was observed when evaluating the generated images with respect to real examples by adopting image generation metrics. We hope that this study and the dataset released will help improving current personalized VL methods.

Supplementary Material

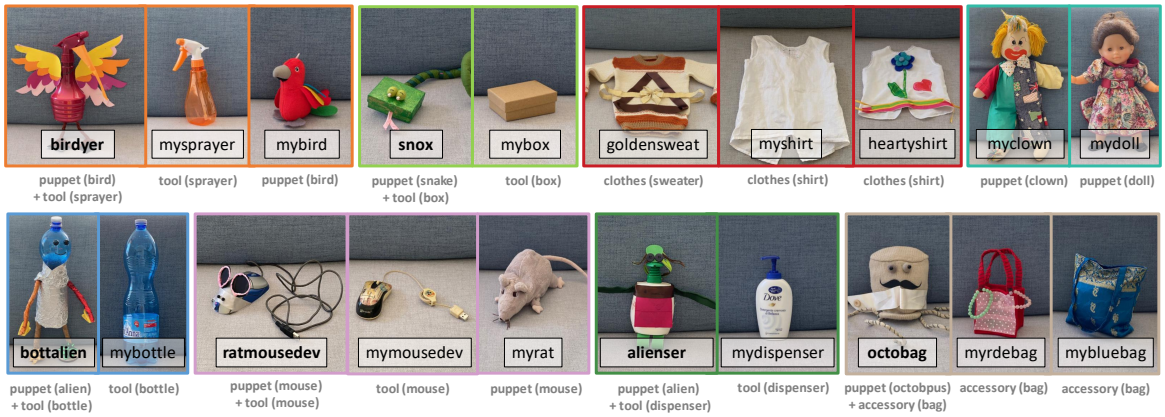


Figure 7.12 **Concepts in ConCon-Chi.** Images from one of the five training set environments. The concepts are grouped together with their hard negatives and the chimeric ones are indicated in (**Bold**).

In this supplementary material we first include additional information about the proposed ConCon-Chi dataset (Section 7.7), then provide additional details about the personalized TIR (Section 7.8) and TIG (Section 7.9) experiments.

7.7 Concept-Context Chimera Benchmark

We provide additional information about the concepts (Section 7.7.1) and the contexts (Section 7.7.2) in the dataset; then we provide more details about the acquisition and annotation procedure (Section 7.7.3) and the proposed benchmark splits (Section 7.7.4). Finally we provide examples of queries/prompts and associated GT images in the dataset (Section 7.7.5).

7.7.1 Concepts: Pictures and Descriptions

In Figure 7.12 we show all the 20 concepts in ConCon-Chi together with their name and category. Chimeric concepts are indicated with a Bold concept name. Other than the 6 chimeras, 4 other objects are considered "animate" (MYDOLL, MYCLOWN, MYBIRD, MYRAT). This type of concept can combine with contexts that require agency (e.g., "sitting" or "playing") and with inanimate concepts. In Table 7.5 we report the three types of concept descriptions used for the TIR and TIG baselines.

7.7.2 Contexts: Kinds and Environments

Contexts are grouped into nine kinds:

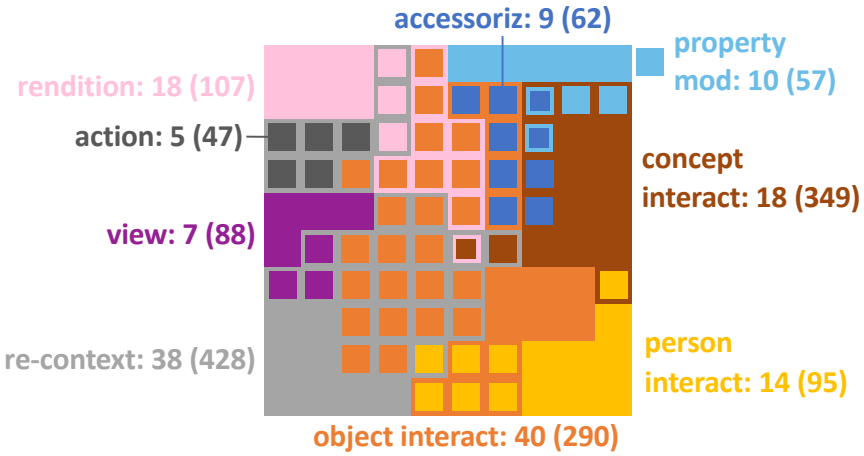
- **re-contextualization**: the concept appears in a different environment;
- **view**: the concept is photographed by a different view (e.g., back, profile, close-up on details);
- **action**: the concept is carrying out an action (e.g., sitting, lying down);
- **accessorization**: the concept is wearing an accessory (e.g., sunglasses, eye-mask)
- **property modification**: the concept is missing some part (e.g., legs, wings, ears) or some part of the concept is applied to another one;
- **interaction**: the concept is interacting with a person, an object or another concept learned in the same way;
- **rendition**: black and white photography, ink or pencil sketch, baby drawing, blurred painting, dark glow-edge effect.

Contexts can be, and in most of the cases are, tagged with multiple kinds. This is visually depicted in Figure 7.13a. From the figure it can be observed that the most frequent kinds are "object interaction" and "recontextualization" which also appear in combination with most of the other kinds. It is also interesting to notice how the 18 contexts tagged as "concept interaction" result in a high number of queries. This is because "concept interaction" contexts allow for numerous concept combinations.

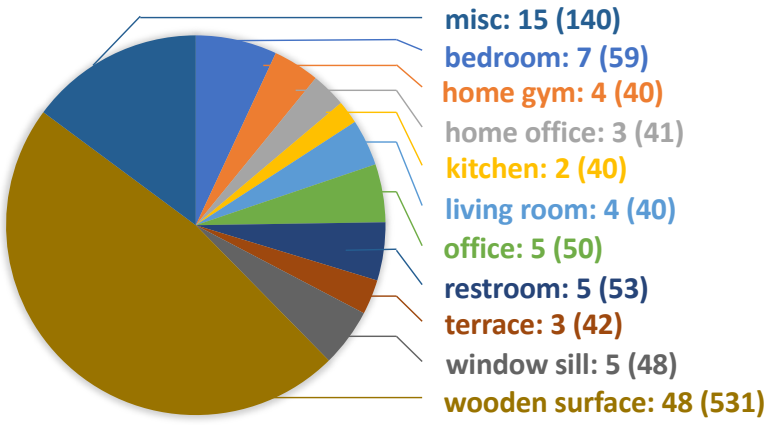
Contexts are also tagged with the environment in which the corresponding ground-truth images were captured. Half of the contexts (48) were captured in a neutral setting (a wooden surface). The other 38 contexts were captured in different environments, coming from two main location types: houses and office spaces. The environment distribution is shown in Figure 7.13b.

7.7.3 Acquisition and Annotation: Procedure Details

Acquisition. We took the images using an iPhone 11 Pro with resolution 4032×3024 and variable focal length. The released pictures were converted to JPEG at a resolution of 1008×756. Part of the renditions was obtained by applying digital effects (ink/pencil sketch, grayscale, blur, dark glow edges) to pictures of 11 concepts manually selected in the image



(a) **Context kinds.** Each cell represents a context and is colored by kind (101 contexts arranged in a 10×10 matrix plus 1 cell at the top-right corner). If a concept belongs to more than a kind, 2 or 3 squares are overlapped in the cell with corresponding colors. The total number of contexts per kind is indicated next to the context name and equals the number of cells with the same color, with the corresponding number of queries in parenthesis.



(b) **Context environments**. The number of contexts per environment is indicated to the left with the corresponding number of queries in parenthesis (and they respectively sum to 101 and 1084).

Figure 7.13

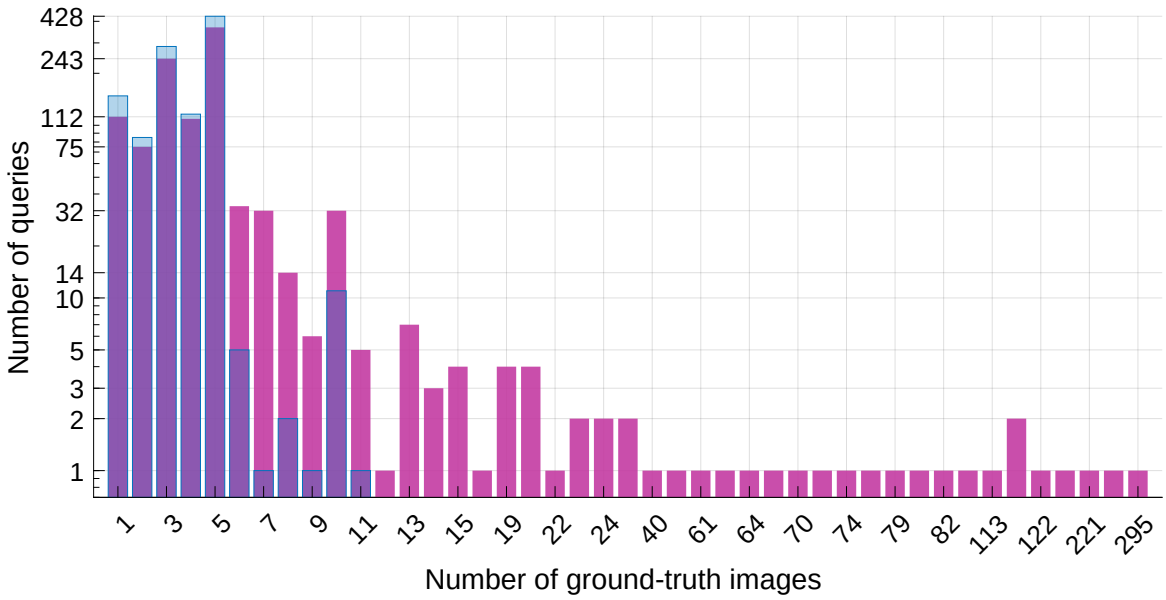


Figure 7.14 **Ground-truth images per query.** The number of ground-truth images that were acquired to represent a given query is reported in Blue (with transparency), the number of overall images that were annotated as ground-truth for a given query is in Magenta.

pool. The remaining part contains actual pen drawings of the concepts sketched on paper and later photographed.

Annotation. The annotation process was divided in two steps. First we used an annotation GUI developed by us to review the images and assign to each of them the correct concepts/context combination. This step produced, for each query, a list of ground-truth (GT) images. The distribution of the number of GTs per query is shown in Figure 7.14 (Blue bars). At this point each of the images is assigned as GT of only one query. However, since some contexts are more general than others, some images needed to be marked as GT for multiple queries. This is necessary to avoid false negatives in the dataset (images that are correct realization of a certain query but are not marked as its GT). To this end, we automatically assigned additional GT images following context overlaps, from more specific to more general queries (e.g., GT images of the query "black and white photography of X resting on a bedside table" are added as GT images for the query "X resting on a bedside table"). Then, the lists of GT images were further extended by manually checking individual GT images of queries which we knew could overlap with others (e.g., we added several images as GT for "X standing on a wooden surface", where the concept was involved in an action or interaction, but in fact was also standing on the surface). In Figure 7.14 we show the resulting number of GT images per query in Magenta.

7.7.4 Benchmark Splits

We report below the splits that we introduce for the personalized TIR and TIG benchmarks (notice that in the main paper we use only the *train* and *test* splits):

- **train**: A set of 5 images per concept each with a different background. For the experiments with $k = 1$ in Section 7.4 we further split the training set into 5 different splits each with one image per concept.
- **val**: A validation set containing 3 concepts (TOOMOUSE, RATMOUSDEV and MYRAT), composed of the (42) queries that contain any of them (but none of the other concepts), set in the OFFICE or WOODEN SURFACE environments. The associated image pool is the union of the ground-truth images of these queries (165 images). Considering the very few-shot nature of the personalization tasks, we do not use the validation set. However, we define and release one since future methods might exploit it.
- **test**: A set that contains all the dataset minus the training images (1084 queries and 4008 images in the pool).
- **test-unseen**: A set from which we removed the queries that contain any of the 3 concepts used for validation (986 queries and 4008 images in the pool).

See Table 7.1 for the further statistics regarding the validation and the test splits.

7.7.5 Dataset Examples

In Figure 7.15 we show 6 examples of caption-image associations present in the dataset, for each of the 6 chimeric concepts (one concept per row). The caption is used as query in TIR and prompt in TIG tasks. In the same way, in Figure 7.16 we report 2 examples of caption-image associations for each of the 14 common concepts (3 concepts per row and 2 in the last row). These two ensembles represent respectively examples of uncommon/novel and common situations, the former more focused on the learning of new-meanings and the latter more focused on the learning of instances of common categories, following related personalization benchmarks.

7.8 Personalized Text-to-Image Retrieval

In this section we provide definition of the TIR metrics adopted in the paper (Section 7.8.1); we report implementation details about the experiment reported in Figure 7.5 (Section 7.8.2);

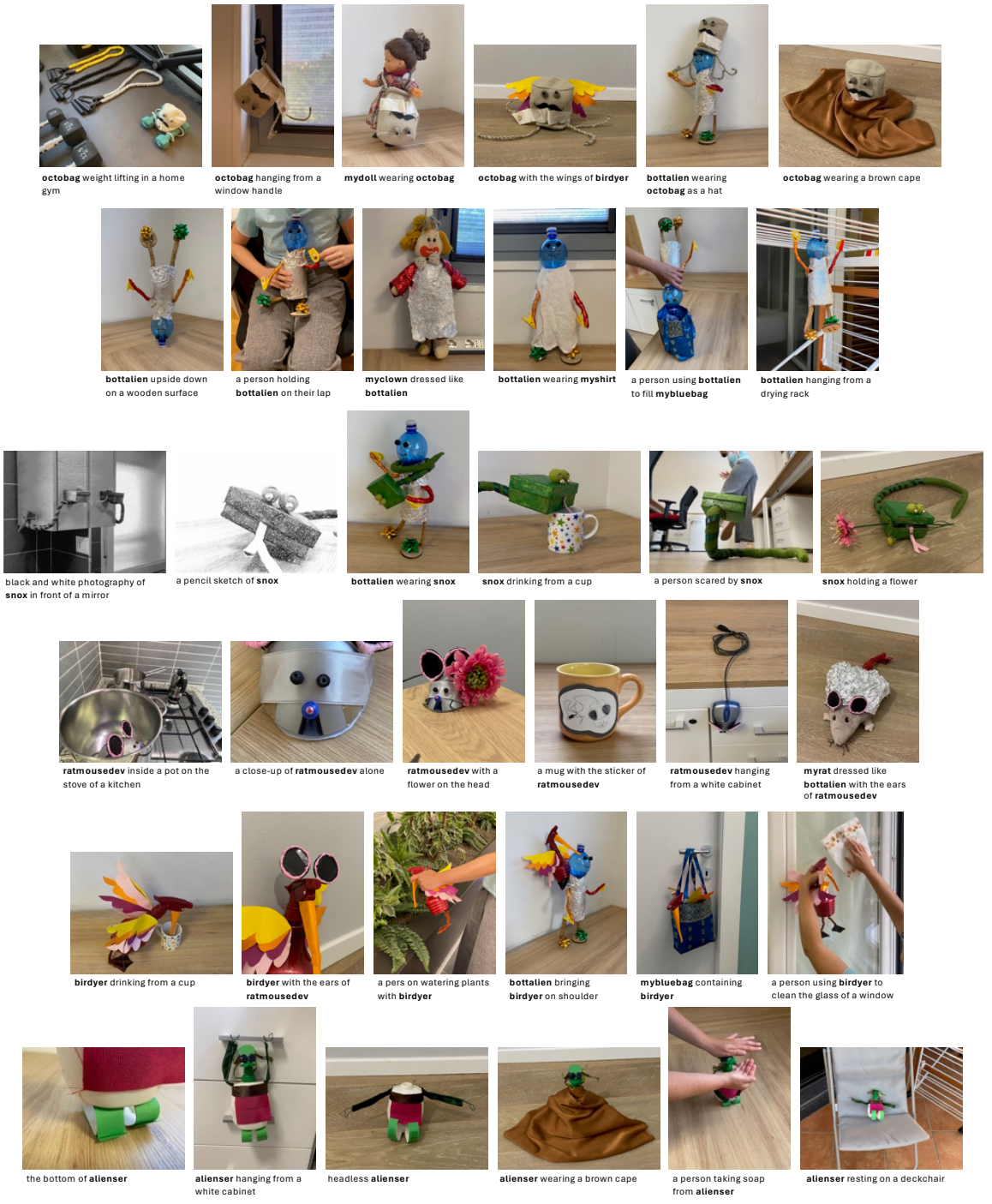


Figure 7.15 **Examples of caption-image associations involving the 6 chimeric concepts.** One concept per row with six examples per concept. The caption is used as query in TIR and prompt in TIG tasks. The concepts are marked in Bold in the captions. Please note that for simplicity we report only one GT image per caption (and only one caption per image).

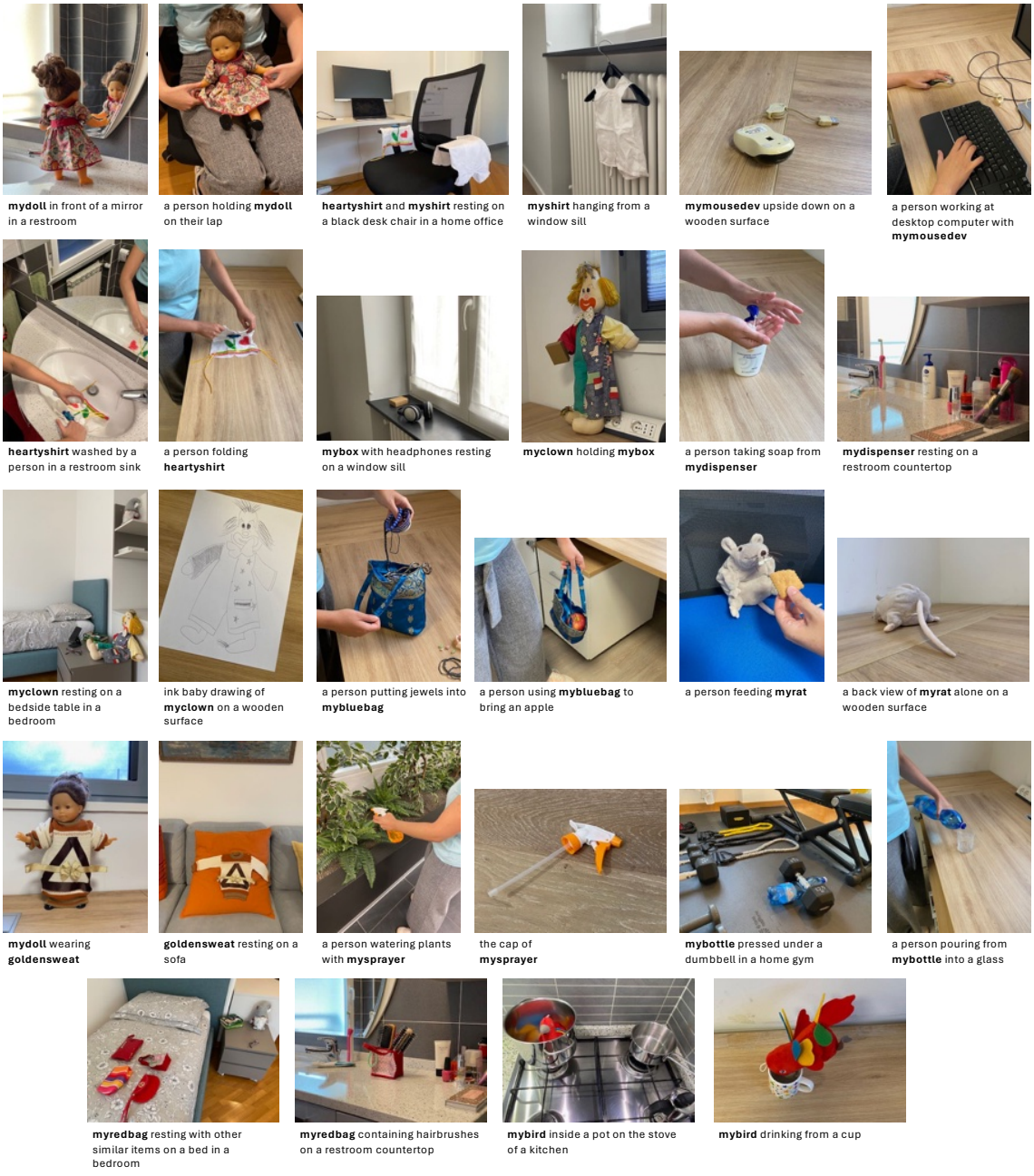


Figure 7.16 **Examples of caption-image associations involving the 14 common concepts.** Three concepts per row and two in the last row. The caption is used as query in TIR and prompt in TIG tasks. The concepts are marked in Bold in the captions. Please note that for simplicity we report only one GT image per caption (and only one caption per image).

we report additional metrics for the TIR benchmark (Section 7.8.3) and additional breakdown of performance (Section 7.8.4) and finally some failure cases (Section 7.8.5).

7.8.1 Metrics Definition

Given a set of queries Q and an image pool I we report below the definition for the adopted metrics:

mean Reciprocal Rank (mRR): the average of the reciprocal rank of the first retrieved ground-truth

$$mRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $rank_i$ refers to the rank of the first retrieved ground-truth for the i-th query.

mean Average Precision at k (mAP@k): the average of the AP (Average Precision) at k, among the queries

$$mAP@k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}AP_i@k$$

with

$$AP_i@k = \frac{1}{min(k,|GTs_i|)}\sum_{m=1}^{k}P_i@m \times rel_i(m)$$

where $P_i@m$ (precision at m) is the fraction of ground-truth images (for the i-th query) among the top m retrieved ones, $rel_i(m)$ is an indicator function which is 1 if the m-th image in I is a ground-truth for the i-th query, and GTs_i is the set of ground-truth images for the i-th query.

mean Average Precision (mAP): it is the mAP@k with $k = |I|$.

Recall at k (R@k): fraction of queries for which the rank of the first retrieved image is smaller than k

$$R@k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\mathbb{1}_{\{rank_i<k\}}$$

where $\mathbb{1}$ is 1 if $rank_i < k$ and 0 otherwise.

7.8.2 Concept and Context Bias: Implementation Details

Below we provide some details about the datasets compared in Figure 7.5.

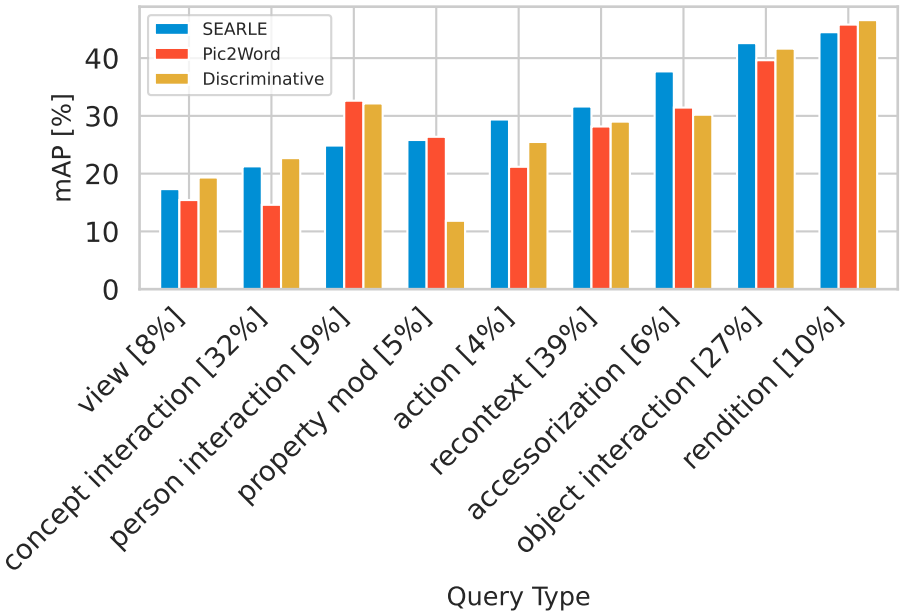


Figure 7.17 **Performance per query kind.** Breakdown of the mAP performance in Table 7.2 per query kind for the two best performing methods and the best text-based baseline.

For FashionIQ, CIRR, CIRCO we used the validation set as the test set annotations are not publicly available. For ConCon-Chi and PerVL DF2 we used the test set.

The PerVL DF2 dataset provides two types of captions: short and detailed. In the paper, the authors define the detailed captions as the ones that describe extensive context about the image and can facilitate retrieval, while the short ones describe less detail and therefore are more ambiguous. We used the detailed captions for our analysis.

The FashionIQ dataset provides images as lists of URLs. However, we found some of the links were broken, making it impossible to download the original dataset. In Table 7.6 we report, for the validation set, the original number of URLs and the number of images that we couldn't download.

Since FashionIQ provides two relative captions for each reference image, as in Baldrati et al. (2023) (Appendix A), we concatenate them.

7.8.3 Additional Benchmark Metrics

To encourage future comparisons on our benchmark, in Table 7.7 we report some additional retrieval metrics for the experiment reported in Table 7.2. In particular we show R@k and mAP@k with k=5, 10.

Figure 7.18 **Retrieval Failure Cases.** First retrieved image for each of the 14 worst performing queries in terms of mAP for SEARLE (Left to Right, Top to Bottom). The part of the query successfully retrieved is highlighted in Green, otherwise in Red.

7.8.4 Additional Analysis of Results

In Figure 7.17 we report the performance per query kind. SEARLE is comparable or better than the baseline across all kinds but "person interaction", where it is outperformed also by Pic2Word, indicating that there may be a limitation specific to its training set. Conversely, the baseline drops heavily on "property modification" and "accessorization".

7.8.5 TIR Failure Cases

We report in Figure 7.18 some failure cases for the top-performing retrieval method, SEARLE. To this end, we sort the queries by decreasing mAP and report the first retrieved image for the last 14 ones. In accordance to the results reported in Table 7.3, on these worst performing queries, SEARLE is almost always able to retrieve the correct concept, but fails to compose it with the correct context.

7.9 Personalized Text-to-Image Generation

In this section we provide implementation information about the Density and Coverage metrics (Section 7.9.1); we then report numbers for the FID metrics (Section 7.9.2) and some failure cases (Section 7.9.3).

7.9.1 Metrics: Implementation Details

To compute Density and Coverage we used the code provided by the authors (Naeem et al., 2020) without substantial modifications[9]. We computed the two metrics on the CLIP embeddings of the real and generated images.

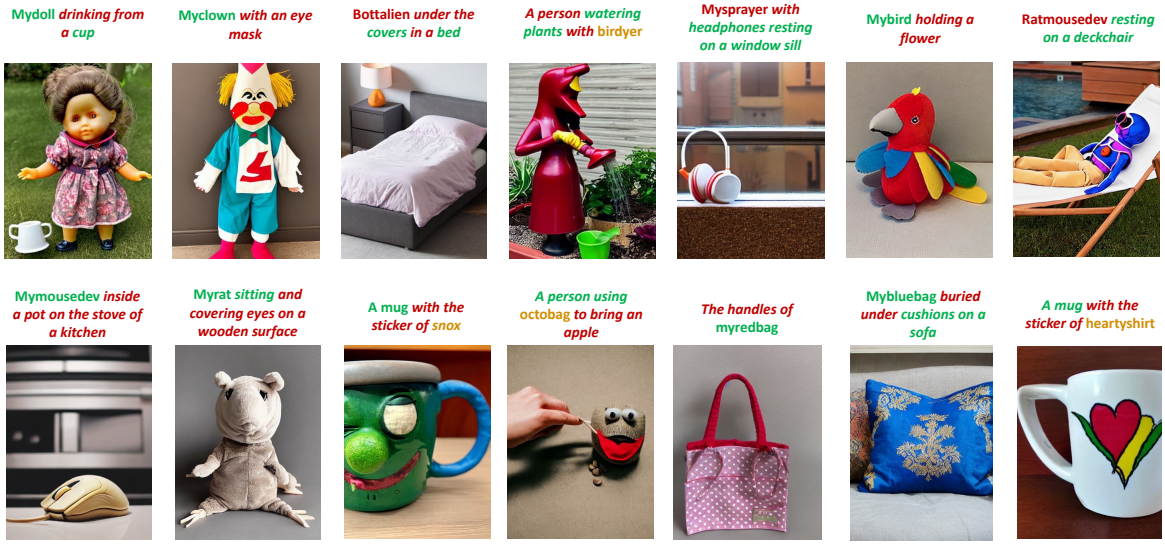


Figure 7.19 **Generation Failure Cases.** Some failure cases for DreamBooth. The part of the query successfully generated is highlighted in Green; the part of the query that seems to be contained in the generated image, but is not completely correct, is in Orange; the part that is not represented in the generated images is in Red. While clearly this type of decision can be subjective, we aimed to provide our judgment as additional information.

7.9.2 Benchmark Results: FID Numbers

In Table 7.8 we report the numbers relative to the scatter plot in Figure 7.10.

7.9.3 TIG Failure Cases

In Figure 7.19 we report failure cases for the TIG benchmark for the DB method. We first observe that sometimes the method represents the elements mentioned in the prompt, but not in the correct relationship (Top Row picture 1, Bottom Row picture 4). In some cases, the elements in the prompt are "semantically merged" (Top Row picture 4, Bottom Row picture 6). In other cases, some element of the prompt is totally missing (Top Row pictures 2, 3, 6;

[9]https://github.com/clovaai/generative-evaluation-prdc

Bottom Row picture 2) or represented inaccurately (Top Row picture 7, Bottom Row pictures 1, 3, 5, 7).

CONCEPT	CATEGORIES	DESCRIPTIONS		
		COARSE (1-TOKEN)	RICH	DISCRIMINATIVE
BIRDYER	**puppet (bird) + tool (sprayer)**	puppet	red plastic sprayer bird puppet with colored wings	bird sprayer puppet
MYBIRD	puppet (bird)	plush	red stuffed bird with colored wings	stuffed bird
MYSPRAYER	tool (sprayer)	plastic container (bottle)	orange transparent plastic sprayer with white cap	sprayer
RATMOUSDEV	**puppet (mouse) + tool (mouse)**	puppet	computer mouse rat puppet with black round ears and cable	rat computer mouse puppet
MYRAT	puppet (mouse)	plush	grey stuffed rat with long pink tail	stuffed rat
MYMOUSEDEV	tool (mouse)	computer mouse (device)	black and beige computer mouse with wheel and cable	computer mouse
BOTTALIEN	**puppet (alien) + tool (bottle)**	puppet	blue alien plastic bottle puppet with aluminum foil dress	alien plastic bottle puppet
MYBOTTLE	tool (bottle)	plastic container (bottle)	blue plastic bottle with label	plastic bottle
ALIENSER	**puppet (alien) + tool (dispenser)**	puppet	green alien soap dispenser puppet with pink dress	alien soap dispenser puppet
MYDISPENSER	tool (dispenser)	plastic container (bottle)	white plastic soap dispenser with blue cap	soap dispenser
SNOX	**puppet (snake) + tool (box)**	puppet	green box headed snake puppet with round eyes and pink tongue	snake box headed puppet
MYBOX	tool (box)	box	cardboard box with lid	box
OCTOBAG	**puppet (octopus) + accessory (bag)**	puppet	beige woolen octopus bag with moustache and scarf	octopus bag puppet
MYREDBAG	accessory (bag)	bag	red square handbag with pois and bracelets	red bag
MYBLUEBAG	accessory (bag)	bag	blue square silk handbag with gold decorations	blue bag
MYDOLL	puppet (doll)	doll	doll with floral dress and brown hair	doll
MYCLOWN	puppet (clown)	puppet	rag clown with yellow hair and jumpsuit	clown
GOLDENSWEAT	clothes (sweater)	sweater	cream wool sweater with brown triangle and gold bow	sweater
HEARTYSHIRT	clothes (shirt)	shirt	white sleeveless baby shirt with red heart and blue flower	baby shirt with flower
MYSHIRT	clothes (shirt)	shirt	plain white baby shirt	white baby shirt

Table 7.5 **Concept descriptions.** Chimeric concepts (**Bold**) are grouped with their hard negative concepts (Dotted lines).

SPLIT	REFERENCE IMAGES	IMAGE POOL
shirt	2038 - 97	6346 - 164
dress	2017 - 156	3817 - 164
toptee	1961 - 94	5373 - 112

Table 7.6 Total number of URLs minus (-) the number of broken ones for each split of FashionIQ.

	Method	mAP[%]	mAP@5[%]	mAP@10[%]	mRR[%]	R@1[%]	R@5[%]	R@10[%]
k=0	*Coarse*	16.83	11.74	13.13	24.21	14.48	33.49	43.63
	Discriminative	*30.16*	*23.91*	*26.47*	*43.16*	*31.92*	***55.63***	***66.14***
	Rich	*27.65*	*21.62*	*23.57*	*40.58*	*29.98*	*51.75*	*62.55*
k=1	PALAVRA	22.56 ± 1.29	17.58 ± 1.08	18.97 ± 1.18	34.39 ± 1.68	24.59 ± 1.94	44.30 ± 1.51	54.85 ± 1.49
	Pic2Word	25.23 ± 1.20	19.52 ± 1.31	21.30 ± 1.18	37.16 ± 1.76	26.35 ± 1.85	48.06 ± 1.91	58.80 ± 1.67
	SEARLE	28.16 ± 0.55	23.02 ± 0.65	24.60 ± 0.57	41.07 ± 0.92	31.16 ± 0.94	51.72 ± 0.71	60.85 ± 1.06
k=5	PALAVRA	23.59	18.65	20.05	35.99	26.75	45.11	55.08
	Pic2Word	26.39	20.67	22.45	38.62	27.68	50.28	60.61
	SEARLE	**30.74**	**25.51**	**27.11**	**43.83**	**33.49**	55.54	63.84

Table 7.7 **Personalized TIR benchmark.** Additional metrics for the results reported in Table 7.2.

	Method	FID_{cpt}	FID_{ctx}
upper bound	concept-only	*92.97*	19.77
	context-only	57.87	*27.92*
	GT images	75.83	25.37
k=0	LDM	70.37	22.35
	SDM	73.35	23.06
k=5	TI (LDM)	70.19	**25.16**
	TI (SDM)	77.25	20.92
	DB (SDM)	**80.42**	23.99

Table 7.8 **Personalized TIG benchmark**: Fidelity-concept/context metrics on ConCon-Chi.

Chapter 8

KDPE: A Kernel Density Estimation Strategy for Diffusion Policy Trajectory Selection

Andrea Rosasco, Federico Ceola, Giulia Pasquale, and Lorenzo Natale

Abstract

Learning robot policies that capture multimodality in the training data has been a long-standing open challenge for behavior cloning. Recent approaches tackle the problem by modeling the conditional action distribution with generative models. One of these approaches is Diffusion Policy, which relies on a diffusion model to denoise random points into robot action trajectories. While achieving state-of-the-art performance, it has two main drawbacks that may lead the robot out of the data distribution during policy execution. First, the stochasticity of the denoising process can highly impact on the quality of generated trajectory of actions. Second, being a supervised learning approach, it can learn data outliers from the dataset used for training. Recent work focuses on mitigating these limitations by combining Diffusion Policy either with large-scale training or with classical behavior cloning algorithms. Instead, we propose KDPE, a Kernel Density Estimation-based strategy that filters out potentially harmful trajectories output of Diffusion Policy while keeping a low test-time computational overhead. For Kernel Density Estimation, we propose a manifold-aware kernel to model a probability density function for actions composed of end-effector Cartesian position, orientation, and gripper state. KDPE overall achieves better performance than Diffusion Policy on simulated single-arm tasks and real robot experiments.

Additional material and code are available on our project page: https://hsp-iit.github.io/KDPE/.

8.1 Introduction

Diffusion Policy (DP) (Chi et al., 2023) has recently emerged as a powerful robotic policy representation due to its capability of handling multimodal behaviors. DP models the robot policy as a Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020), and during policy execution denoises a set of randomly sampled trajectory points into the trajectory that the robot is requested to execute with receding horizon control (Mayne and Michalska, 1988). While enabling the robot to capture multimodality in the demonstrations used for training, the choice of the random points to initialize the denoising process performed at inference can lead the robot to execute different trajectories, given the same observation. This may be problematic if the sampled trajectory is an outlier with respect to modes that are most represented in the training data, leading the robot into a state which may be out of the distribution of the demonstrations dataset.

To overcome this limitation, we propose KDPE: a strategy to sample trajectories that are representative of the modes learned by the policy. We propose to compute a set of N trajectories, by performing in parallel N denoising processes from N different starting noise samples conditioned on the same current observation. We then estimate the Probability Density Function (PDF) on the last actions in the trajectories generated by Diffusion Policy, using Kernel Density Estimation (KDE), and select the trajectory associated to the action with the highest density.

The actions we consider are composed of end-effector pose and gripper state. Modeling a KDE on a population of actions requires a kernel function that relates end-effector position, orientation and gripper state. While probability distributions on end-effector poses have previously been proposed (Barfoot and Furgale, 2014; Gilitschenski et al., 2014), none of them integrate the gripper state. We modify the probability distribution introduced in Barfoot and Furgale (2014) to handle the gripper state component.

In principle, the multimodality of the trajectories predicted by DP can be modeled with non-parametric clustering algorithms (Comaniciu and Meer, 2002). However, these approaches require non-negligible convergence time when applied for outlier rejection (Ester et al., 1996) in closed-loop control tasks. KDPE, instead, manages to filter and sample among a subset of the trajectories predicted by DP with a significantly lower computational overhead, making it suitable for the target visuomotor closed-loop control tasks. While KDPE uses DP for trajectory generation, the core idea of sampling multiple trajectories and filtering them through KDE can be applied to any probabilistic generative policy.

We evaluate KDPE against DP on the four RoboMimic (Mandlekar et al., 2022) single-arm tasks used in Chi et al. (2023) to benchmark DP, and on three tasks from the MimicGen benchmark (Mandlekar et al., 2023). We train DP on these tasks, and evaluate the models with KDPE, achieving an overall improvement in terms of average success rate. We then evaluate KDPE's robustness to visual perturbations by changing the color of an object in the task environments and show that KDPE maintains performance closer to the non-shifted setting than DP.

Finally, using a Franka Emika Panda (Haddadin et al., 2022) manipulator, we test KDPE on three real-world tasks: *PickPlush*, a tabletop picking task where a plush has to be grasped and lifted by the robot, *CubeSort*, a multi-step multimodal task where the robot has to pick two cubes and place them in the cup of the corresponding color, and *CoffeeMaking*, a fine-grained multi-step task where the robot has to load a coffee machine with a pod. Overall, KDPE outperforms DP in terms of success rate, and qualitatively shows smoother behavior.

In summary, the contributions of the paper are:

- **KDPE**: a KDE-based approach for selection of DP action trajectories that are representative of the different action modalities.
- A **manifold-aware kernel for KDE** on robotic actions composed of Euclidean end-effector position and gripper aperture, and non-Euclidean end-effector orientation.
- An **extended quantitative evaluation of KDPE** on seven simulated tasks from two different benchmarks and on three real-world tasks.
- A **trajectory visualizer** for qualitative analysis of manipulation policies.

8.2 Related Work

There is a huge effort in the robotics community to create generalist robot policies (Bousmalis et al., 2024; Ghosh et al., 2024; Kim et al., 2024) from large-scale datasets (Khazatsky et al., 2024; Open X-Embodiment Collaboration, 2024), either with large transformer-based models (Black et al., 2025a,b; Bousmalis et al., 2024; Brohan et al., 2023; Ghosh et al., 2024), or parameterizing robot policies as language tokens (Kim et al., 2024; Team et al., 2025; Zitkovich et al., 2023). However, for single tasks, there is evidence (Kim et al., 2024) that DP (Chi et al., 2023) achieves better performance than large generalist models (Ghosh et al., 2024; Kim et al., 2024).

DP (Chi et al., 2023), together with its flow matching variant (Zhang et al., 2023), is currently one of the most widely used approaches for behavior cloning for robotic manipulation tasks. Its policy parameterization as a Denoising Diffusion Probabilistic Model (DDPM) allows to model the multimodality in the training data. Several recent works extend the original DP problem formulation to learn goal-conditioned (Reuss et al., 2023), or language-conditioned (Dasari et al., 2024; Ha et al., 2023) tasks. Octo (Ghosh et al., 2024), instead, uses DP to parameterize a generalist robot manipulation policy on top of a large transformer-based modular backbone for transferability across tasks and embodiments. Approaches like the one presented in Zhang et al. (2024) combine DP with more classical behavior cloning algorithms like DAgger (Ross et al., 2011) to reduce the compounding errors typical of pure imitation learning algorithms.

Recent work has explored failure detection and mitigation strategies to enhance the reliability of robot policies. Sentinel (Agia et al., 2025) identifies potential failures by analyzing the consistency of overlapping trajectory segments sampled at different timesteps, and leverages vision-language models to detect misalignment between actions and visual context. While effective, such methods often detect failures after the policy has already entered an out-of-distribution state, making recovery challenging. RoboMD (Sagar et al., 2025) addresses robustness by systematically identifying failure modes through DRL-guided exploration across diverse environmental conditions. RoboMD fine-tunes policies on these identified failure modes. This *train-deploy-analyze-retrain* pipeline, however, can be impractical for real-world deployment. V-GPS (Nakamoto et al., 2024) improves robot policies at inference time by using a value function network to rank actions. However, this requires to train a value function network with offline Reinforcement Learning and design a reward function. In contrast, KDPE operates at inference time without requiring any additional training. By leveraging the inherent diversity of DP's trajectory sampling process, KDPE mitigates failures proactively, selecting trajectories that align with the statistical modes of the learned distribution, rather than relying on post-hoc detection or costly retraining.

A growing body of literature in Natural Language Processing studies *inference or test-time scaling* methods that select one among multiple output samples generated (e.g. by a Large Language Model) to improve the performance of the trained model (Brown et al., 2024; Snell et al., 2025; Wu et al., 2024). Furthermore, there is recent research interest in the application of inference-time scaling to, e.g., text-to-image diffusion models (Li et al., 2025; Ma et al., 2025). In Xie et al. (2025) the effectiveness of the easiest *best-of-N sampling* (generation of a high number of random samples and selection of the best one by using an external evaluator) is proved experimentally. To the best of our knowledge, however, KDPE

is the first method that improves DP at inference time based on statistical properties of the trajectories distribution. Furthermore, there is recent research interest in the application of inference-time scaling to, e.g., text-to-image diffusion models (Li et al., 2025; Ma et al., 2025). In Xie et al. (2025) the effectiveness of the easiest *best-of-N sampling* (generation of a high number of random samples and selection of the best one by using an external evaluator) is proved experimentally. To the best of our knowledge, however, KDPE is the first method that improves DP at inference time based on statistical properties of the trajectories distribution.

8.3 Methodology

8.3.1 Diffusion Policy

DP adapts DDPM, a generative diffusion model which is commonly used for image generation, to behavior cloning for robotic manipulation tasks in a receding horizon fashion. DP, given an observation of the environment, outputs action trajectories of shape $T \times D$, where T is the number of time-steps and D is the action dimensionality. These trajectories are generated through a denoising process that progressively refines Gaussian noise by iteratively subtracting a learned gradient field. This process allows the policy to model multimodal trajectory distributions and maintain the multimodality of task demonstrations. The denoising process to generate a trajectory is defined as:

$$\mathbf{A}^{k-1} = \alpha(\mathbf{A}^k - \gamma\varepsilon_\theta(\mathbf{A}^k, k) + \mathcal{N}(0, \sigma^2 I)), \tag{8.1}$$

where ε_θ is the noise prediction network with parameters θ and $\mathbf{A}^k$ is a noisy sample going through a denoising step. This process is repeated for K steps, starting with $\mathbf{A}^K$ as randomly sampled Gaussian noise, to output the trajectory $\mathbf{A}^0$. During training, a trajectory is sampled from the dataset and is perturbed by adding the appropriate amount of noise corresponding to a random denoising step k. The noise prediction network ε_θ takes as input the noisy sample and is optimized to predict the noise ε^k that has been added to the ground-truth trajectory $\mathbf{A}^0$ with the following loss function:

$$\mathscr{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathbf{A}^0 + \varepsilon^k, k)). \tag{8.2}$$

8.3.2 KDPE

KDPE enhances Diffusion Policy by scoring the population of trajectories via Kernel Density Estimation (KDE) (Parzen, 1962) and selecting the best using *best-of-N sampling*. Our approach combines a manifold-sensitive kernel with multi-hypothesis sampling from the diffusion process. Given an observation $\mathbf{o}_t$, we sample N independent and identically distributed action trajectories $\{\mathbf{A}_i\}_{i=1}^{N} \sim p_{\boldsymbol{\theta}}(\mathbf{A}|\mathbf{o}_t)$, where $\mathbf{A}_i \in \mathbb{R}^{T\times D}$, and estimate the PDF of the last action of the trajectories, hereinafter defined as $\mathbf{a}_i \in \mathbb{R}^D$, via KDE. This allows to obtain the probability density of every action and use it to discard outlier trajectories.

To model the PDF, KDE requires a unified kernel over all the action components representing end-effector position and orientation, and gripper aperture. While the multivariate Gaussian kernel is a natural choice for Euclidean domains, it cannot directly handle data lying on different manifolds such as rotations in $\mathrm{SO}(3)$, being a non-Euclidean space.

Manifold-Aware Kernel Density Estimation We propose a modification of the probability distribution presented in Barfoot and Furgale (2014) to handle the pose of the end-effector together with the gripper state. Although DP outputs orientations in the 6D matrix representation introduced in Zhou et al. (2019), we denote by $\mathbf{R} \in \mathrm{SO}(3)$ their corresponding rotation matrices and use them in the following for clarity. Our kernel function is defined by the equation

$$k(\mathbf{a}_i, \mathbf{a}_j) = \frac{1}{\sqrt{(2\pi)^D |\mathbf{H}|}} \exp\left(-\frac{1}{2}\boldsymbol{\Delta}_{ij}^{\top}\mathbf{H}^{-1}\boldsymbol{\Delta}_{ij}\right), \tag{8.3}$$

where $\mathbf{a}_i = [\mathbf{t}_i; \mathbf{R}_i; \mathbf{g}_i]$ and $\mathbf{a}_j = [\mathbf{t}_j; \mathbf{R}_j; \mathbf{g}_j]$ are actions composed of position, rotation, and gripper aperture components. $\mathbf{H}$ is the covariance matrix defined as $\mathbf{H} = \mathrm{diag}\big(\sigma_{\mathrm{pos}}^2\mathbf{I}_3,\, \sigma_{\mathrm{rot}}^2\mathbf{I}_3,\, \sigma_{\mathrm{grip}}^2\mathbf{I}_1\big)$, and $\boldsymbol{\Delta}_{ij} \in \mathbb{R}^D$ represents the difference between manifold-specific components defined as:

$$\boldsymbol{\Delta}_{ij} = [\underbrace{\mathbf{t}_i - \mathbf{t}_j}_{\text{Euclidean}}; \underbrace{\log(\mathbf{R}_j^T\mathbf{R}_i)^{\vee}}_{\mathrm{SO}(3)}; \underbrace{\mathbf{g}_i - \mathbf{g}_j}_{\text{Euclidean}}] \tag{8.4}$$

For position and gripper components, standard Euclidean differences suffice. For rotations represented as rotation matrices $\mathbf{R}_i, \mathbf{R}_j \in \mathrm{SO}(3)$, we compute the transformation from $\mathbf{R}_j$ to $\mathbf{R}_i$ and convert it to its axis-angle representation using the Lie group logarithm map $\log : \mathrm{SO}(3) \to \mathfrak{so}(3)$ followed by the vee operator $(\cdot)^{\vee} : \mathfrak{so}(3) \to \mathbb{R}^3$ to obtain a representation in the tangent space. From the definitions above, we can rewrite $-\frac{1}{2}\boldsymbol{\Delta}_{ij}^{\top}\mathbf{H}^{-1}\boldsymbol{\Delta}_{ij}$ as

$$-\frac{1}{2}\left(\frac{\|\mathbf{t}_i - \mathbf{t}_j\|^2}{\sigma_{\mathrm{pos}}^2} + \frac{\|\log(\mathbf{R}_j^{\top}\mathbf{R}_i)^{\vee}\|^2}{\sigma_{\mathrm{rot}}^2} + \frac{(\mathbf{g}_i - \mathbf{g}_j)^2}{\sigma_{\mathrm{grip}}^2}\right), \tag{8.5}$$

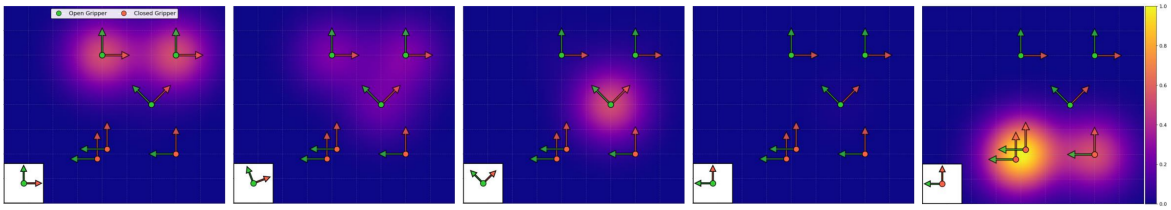


Figure 8.1 Visualization of the PDF estimated via KDE with the proposed manifold-aware kernel. We perform KDE on a population of 6 planar end-effector actions represented as reference frames in the plots. Three of them represent open grippers (green circle), while the other three represent closed grippers (red circle). The color of each point in the heatmaps represents the density value of an action at the corresponding 2D location, that has orientation and gripper state showed by the indicator frame in the white square of each plot. From left to right we vary the rotation of the indicator frame from 0 to 90 degrees and observe that the densities returned by KDE at different locations vary accordingly, spiking when the probed actions are close to the ones used for PDF modeling. The plots show how KDE correctly handles multimodality by providing the highest density values for the most well represented samples. The two rightmost plots show how the gripper state is correctly handled by the KDE.

where $\| \log(\mathbf{R}_j^\top \mathbf{R}_i)^\vee \|^2$ is the geodesic distance representing the minimal rotation angle θ_{ij} required to align $\mathbf{R}_i$ and $\mathbf{R}_j$:

$$d_{SO(3)}(\mathbf{R}_i, \mathbf{R}_j) = \| \log(\mathbf{R}_j^\top \mathbf{R}_i)^\vee \|^2 = \| \boldsymbol{\theta}_{ij} \| = \theta_{ij}.$$

In Figure 8.1, we show how the proposed manifold-aware kernel allows to model PDFs for actions comprising end-effector position and orientation, and gripper aperture.

Density Estimation and Action Selection We model the PDF underlying the N actions $\{\mathbf{a}_i\}_{i=1}^N$ predicted by DP with KDE, by combining kernel densities across all actions. Specifically, we compute the density of a generic action $\tilde{\mathbf{a}}$ as:

$$\rho(\tilde{\mathbf{a}}) = \frac{1}{N} \sum_{i=1}^{N} k(\tilde{\mathbf{a}}, \mathbf{a}_i) \tag{8.6}$$

Finally, we compute densities $\rho_i = \rho(\mathbf{a}_i)$ for each of the N final actions predicted by DP and we select the trajectory $\mathbf{A}_i$ corresponding to the action $\mathbf{a}_i$ with the highest ρ_i.

8.3.3 Tr-KDPE

Tr-KDPE (Trajectory-KDPE) is a modification of the KDPE algorithm that estimates the probability density function (PDF) over a population of complete action trajectories. We

recall that KDPE instead selects action trajectories based only on the density associated to the final action. Therefore, Tr-KDPE mitigates the risk of favoring outlier trajectories that, while ending in an in-distribution state for KDPE, may overall represent out-of-distribution trajectories.

To model the joint PDF $p(\mathbf{a}_1, ..., \mathbf{a}_T)$ of the entire action sequence, Tr-KDPE uses the Markov assumption to decompose this joint probability into a product of conditional probabilities: $p(\mathbf{a}_1, ..., \mathbf{a}_T) = p(\mathbf{a}_1) \prod_{t=2}^{T} p(\mathbf{a}_t | \mathbf{a}_{t-1})$. The estimation relies on modeling the transition probabilities $p(\mathbf{a}_t | \mathbf{a}_{t-1})$ with conditional Kernel Density Estimation (Hyndman et al., 1996):

$$\hat{f}(y|x) = \frac{\hat{g}(x,y)}{\hat{h}(x)}, \tag{8.7}$$

where $\hat{g}(x,y)$ is the KDE of the joint density $g(x,y)$

$$\hat{g}(x,y) = \frac{1}{N} \sum_{j=1}^{N} k\left(x, x_j\right) k\left(y, y_j\right) \tag{8.8}$$

and $\hat{h}(x)$ is the KDE of the marginal density $h(x)$

$$\hat{h}(x) = \frac{1}{N} \sum_{j=1}^{N} k\left(x, x_j\right). \tag{8.9}$$

Therefore, we can compute the kernel density estimator of $p(\mathbf{a}_t | \mathbf{a}_{t-1})$ as:

$$\rho\left(\mathbf{a}_t | \mathbf{a}_{t-1}\right) = \frac{\hat{g}(\mathbf{a}_{t-1}, \mathbf{a}_t)}{\hat{h}(\mathbf{a}_{t-1})}. \tag{8.10}$$

Consequently, estimating the full trajectory density $p(\mathbf{a}_1, ..., \mathbf{a}_T)$ involves computing one standard KDE for $p(\mathbf{a}_1)$ and $T-1$ conditional KDEs for the terms $t = 2, ..., T$.

8.4 Experimental Setup

8.4.1 Simulated Environments

We evaluate KDPE on seven tasks from two different benchmarks: RoboMimic (Mandlekar et al., 2021) and MimicGen (Mandlekar et al., 2023). We consider four RoboMimic tasks: *Lift*, *Can*, *Square*, and *ToolHang* (illustrated in Fig. 8.2). This subset of RoboMimic tasks corresponds to the single-arm image-based tasks used to evaluate DP in (Chi et al., 2023).

RoboMimic provides two datasets for *Lift*, *Can*, and *Square*: one collected by proficient human operators (*ph*) and another by a mix of proficient and non-proficient operators (*mh*), resulting in lower data quality for *mh*. For *ToolHang*, only the *ph* dataset is available. We also test KDPE on three MimicGen tasks: *Coffee*, *Stack Three* (*Stack* for conciseness) and *Three Piece Assembly* (*Assembly*). We chose this benchmark because it shares the same structure as RoboMimic, making the integration with the evaluation framework seamless and allowing us to test KDPE on additional challenging tasks.

To assess the robustness of KDPE under visual domain shift, we compare its performance with DP on perturbed variations of the original environments where we slightly modify the color of an object. Specifically, we remove the original texture from specific objects and set their new color as the average value of the original texture with a decreased 10% lightness value under the HSL color scheme. The specific objects modified for this experiment are shown in Section 8.8.

8.4.2 Comparison with Diffusion Policy

To compare KDPE with Diffusion Policy (DP), we train DP for $80k$ training steps. For each experimental setting, we rollout the methods for 100 random resets of the environment. The random sequence of environment initializations and the DDPM noise schedules are kept fixed across experiments. During policy rollout, we sample a population of 100 trajectories at every inference step, and one trajectory is selected differently for each method. The DP baseline uniformly samples the output trajectory from the population. For KDPE, we perform KDE on the eighth actions of the population of trajectories, as eight is the action execution horizon used in DP. If the selection method is non-deterministic, as it is the case for DP, we run the full set of rollouts three times and report the average result.

Additionally to the comparison with DP, we report performance on two modified versions of KDPE: KDPE-OOD and Tr-KDPE. KDPE-OOD is a modification of KDPE, where we choose the trajectory associated to the least represented action, i.e., the action with the minimum density $\{\rho_i\}_{i=1}^{N}$ (see Section 8.3.2). We evaluate this method to further support the need for outliers rejection in the output trajectories of DP. Tr-KDPE, instead, is a modification of KDPE which uses conditional KDE and a first order Markovian assumption to estimate the PDF of the population of whole trajectories. We describe Tr-KDPE in Section 8.3.3.

We present results for both the CNN-based (DP-C, KDPE-C, KDPE-OOD-C, Tr-KDPE-C) and the Transformer-based (DP-T, KDPE-T, KDPE-OOD-T, Tr-KDPE-T) models. For

Figure 8.2 RoboMimic (*Lift*, *Can*, *Square* and *ToolHang*) and MimicGen (*Coffee*, *Stack* and *Assembly*) tasks considered for KDPE's evaluation.

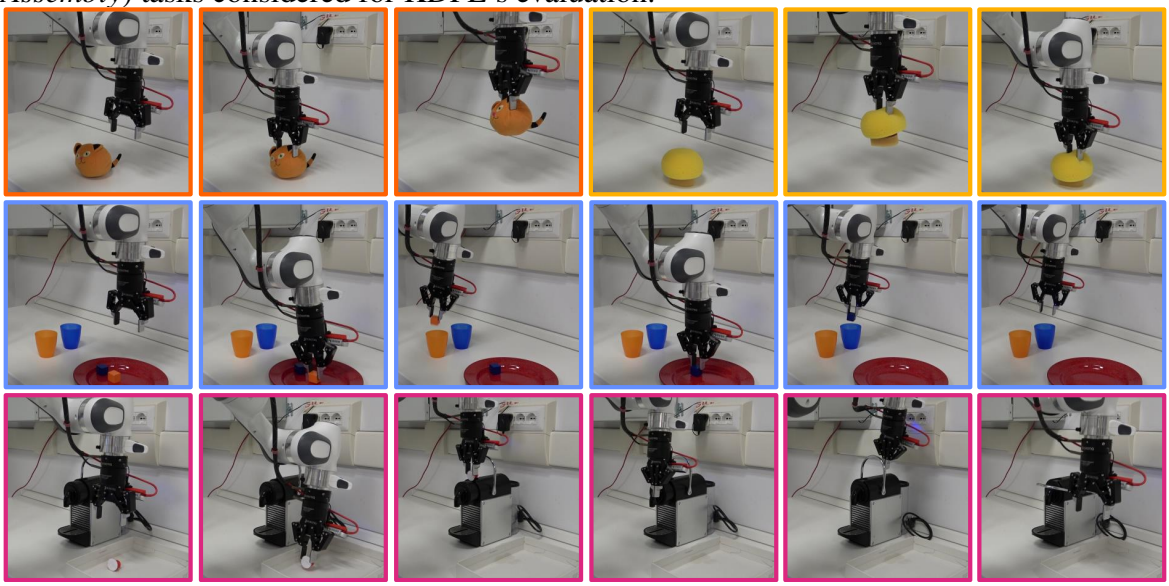


Figure 8.3 KDPE autonomously executing the real-robot tasks: *PickPlush* (orange border) and its variant *PickSponge* (yellow), *CubeSort* (blue) and *CoffeeMaking* (purple).

all methods, we use the same hyperparameters, e.g. KDE kernel bandwidths, reported in Section 8.9.

8.4.3 Real Robot

We collect datasets to train the policies by using a *Meta Quest 3* VR headset to teleoperate a *Franka Emika Panda* robotic arm equipped with a *Robotiq 2f-85* gripper. We provide visual RGB observations to the robot from an *Intel(R) RealSense D415* mounted on the wrist of the robot and an external *Intel(R) RealSense D405* (Figure 8.3). We collect 50 demonstrations for *PickPlush*, 135 demonstrations for *CubeSort* and 200 demonstrations for *CoffeeMaking*.

We first test DP and KDPE for 50 episodes on *PickPlush* (Figure 8.3), a picking task that consists of grasping and lifting a cat plush from the table. Additionally, mimicking the visual domain shift experiment performed in simulation, we test the same checkpoints on a version of *PickPlush* where the plush is replaced with a yellow sponge, named *PickSponge*. Then, we compare DP and KDPE for 100 episodes on *CubeSort*. The robot is required to grasp a blue and an orange cube from a red plate and place them in the cups of the corresponding

	Lift		Can		*Square*		*ToolHang*	*Coffee*	*Stack*	*Assembly*	Average
	ph	*mh*	*ph*	*mh*	*ph*	*mh*	*ph*				
DP-C	**100**	**100**	96.7	94.0	**93.0**	83.0	62.0	85.0	80.7	54.3	84.9
KDPE-OOD-C	**100**	**100**	96.0	84.0	83.0	49.0	7.0	85.0	72.0	40.0	71.6
Tr-KDPE-C	**100**	**100**	97.0	94.0	90.0	78.0	68.0	76.0	78.0	**61.0**	84.2
KDPE-C	**100**	**100**	**98.0**	**96.0**	92.0	**86.0**	**76.0**	**88.0**	**85.0**	**61.0**	**88.2**
DP-T	**100**	**100**	96.0	90.0	84.0	73.0	61.3	91.7	73.7	**23.3**	79.3
KDPE-OOD-T	**100**	99.0	95.0	91.0	84.0	65.0	2.0	**93.0**	61.0	23.0	71.3
Tr-KDPE-T	**100**	99.0	93.0	92.0	81.0	81.0	62.0	86.0	**74.0**	18.0	78.6
KDPE-T	**100**	**100**	**97.0**	**92.0**	**88.0**	**83.0**	**66.0**	91.0	72.0	23.0	**81.2**

Table 8.1 Average success rate (%) of DP, KDPE-OOD, Tr-KDPE and KDPE on RoboMimic and MimicGen. The last column reports the average success rate over all the tasks.

colors. This requires the policy to learn a task with a high degree of multimodality both in the order in which the cubes are picked and in the way they are manipulated. In the task demonstrations, the two cubes are grasped in a random sequence and, during model evaluation, the cubes are randomly tossed on the plate at the beginning of each episode. Finally, we evaluate for 10 episodes the two methods on the *CoffeMaking* task, a real-world experiment to understand the impact of KDPE in a realistic scenario requiring long-horizon planning capabilities and considerable precision. The *CoffeeMaking* task requires the robot to complete the following steps: picking up a coffee pod, inserting the pod in the coffee machine with poor visual conditions, pushing the front part of the coffee machine to close it, and pulling down the metal handle. Some of these steps require high precision, e.g. for pod insertion, as it can be noticed from Figure 8.3, where the task is autonomously performed by KDPE.

Following the methodology described in Section 8.4.2, we train the CNN-based version of the DP model for 80k steps from images of size 480×480 and we evaluate DP and KDPE on the same checkpoint, using DDIM (Song et al., 2021) as noise scheduler to speed-up inference.

8.5 Results

8.5.1 Benchmark on RoboMimic and MimicGen Tasks

Results in the benchmark reported in Table 8.1 show that KDPE outperforms DP on both architectures, and that KDPE-C is overall the best-performing method, achieving an average success rate of 88.2%. KDPE improves DP by 3.3% when using the CNN backbone, and by

	Lift		*Can*		*Square*		*ToolHang*	*Coffee*	*Stack*	*Assembly*	Average
	ph	*mh*	*ph*	*mh*	*ph*	*mh*	*ph*				
DP-C	**100**	99.6	92.3	**91.3**	93.7	79.7	59.0	92.3	75.3	44.7	82.8
KDPE-C	**100**	**100**	**94.0**	90.0	**94.0**	**90.0**	**69.0**	**97.0**	**88.0**	**50.0**	**87.2**
DP-T	98.7	**100**	89.0	84.0	**79.3**	68.3	58.0	**73.3**	67.7	15.6	73.4
KDPE-T	**99.0**	**100**	**90.0**	**85.0**	75.0	**77.0**	**62.0**	72.0	**68.0**	**20.0**	**74.8**

Table 8.2 Success rate (%) of DP and KDPE under object color perturbation.

1.9% with the Transformer-based model. These gaps are even more pronounced (4.2% and 2.4%) if we exclude from the analysis the *Lift* task which saturates to a success rate of 100% for both methods. It is also worth noting that, in the three experiments where DP outperforms KDPE, the performance gap is limited and the success rate of KDPE is always within the standard deviation over the three trials of DP (see Section 8.10).

KDPE achieves a larger performance improvement w.r.t. DP on precision tasks as *ToolHang*, and on tasks learned from lower quality data (*mh*). The first observation indicates that the success rate of high-precision tasks is more influenced by outlier trajectories. The second, paired with the intuition that policies trained on noisy data learn to generate the outliers contained in the dataset, underlines the importance of our filtering mechanism. These findings are further supported by the performance of KDPE-OOD that, in a specular way, experiences its largest drops in performance on the tasks where KDPE performs best. The trajectory-based version of KDPE achieves similar performance to DP, resulting in no relevant improvement on either architecture. We attribute this to the dimensionality of Tr-KDPE's kernel. In fact, by assuming each action to be conditioned on the previous one, the dimension of the KDE samples doubles, becoming $2D = 20$. This might lead to curse of dimensionality and poor characterization of the population's distribution. While the population size could be increased, it is important to notice that the required sample size increases exponentially with respect to dimensionality. This would lead to an exponential increase of the computational cost.

8.5.2 Analysis Under Visual Environment Perturbations

To study the robustness of DP and KDPE to domain shift, we perturb the model observations through the object color modification presented in Section 8.4.1, and measure the success rate on the same set of RoboMimic and MimicGen tasks considered in Section 8.5.1. Results in Table 8.2 show that, on average, KDPE outperforms DP with both CNN and Transformer-based models by a similar margin to that presented in Table 8.1 for the benchmark experiments

	PickPlush	*PickSponge*	*CubeSort*	*CoffeeMaking*
DP-C	90	88	41	60
KDPE-C	**96**	**90**	**44**	**70**

Table 8.3 Success rate (%) of DP-C and KDPE-C on the real-world tasks. We test *PickPlush* and *PickSponge* for 50 episodes, *CubeSort* for 100 episodes and *CoffeeMaking* for 10 episodes.

in the unperturbed setting. This result further supports the effectiveness of KDPE in filtering out trajectory outliers, even in noisier settings.

8.5.3 Real Robot Results

We compare KDPE to DP on three real-world tasks to study whether the improvement in performance observed in the simulated tasks translates to better performance on the real robot. Results in Table 8.3 show that in the four experiments KDPE outperforms DP in terms of success rate. Moreover, for all the tasks, we noticed a smoother behavior of robot end-effector and gripper aperture. This suggests that KDPE allows to choose trajectories, representing task modalities, more consistently.

PickPlush We evaluate performance for 50 episodes on the two different task configurations described in Section 8.4.3. We initialize each episode with the plush in a random position. However, when one of the two methods fails to grasp the object, we perform a rollout of the other method starting from the same object position. KDPE solves the task with the orange plush, *PickPlush*, 48 times, while DP 45 (96% vs. 90% success rate), being in line with the KDPE-C results in Table 8.1. In the three failure cases of DP, the choice of a suitable trajectory with KDPE has been critical to solve the task. We also test the same model (trained only to grasp the orange plush) with the yellow sponge (*PickSponge*). Similarly to the experiments under object color perturbation presented in Section 8.5.2, KDPE outperforms DP, but the gap in performance w.r.t. *PickPlush* reduces. This may indicate that, on the failed episodes with the yellow sponge, DP (and therefore KDPE) could not predict suitable in-distribution trajectories.

CubeSort We evaluate the two algorithms for 100 episodes, by randomly tossing the two cubes on the plate at the beginning of each episode. Differently from *PickPlush*, we could not test the methods on the same object positions, as the failure state of the task is often partial (i.e., only one cube is positioned in the matching cup). KDPE achieves a higher success rate (3% higher) which reflects the results observed in simulation for KDPE-C, e.g. on the *Stack* task which requires to manipulate cubes of a similar size.

CoffeeMaking We report performance for the *CoffeeMaking* task for 10 episodes. We noticed that, for this task, the initial position of the pod on the table has a huge impact on the final success of the task. Therefore, for a fair comparison, we tested both methods with the same pod initializations. KDPE managed to solve the task seven times, while DP six.

8.5.4 Inference Time

We measure the inference time of DP-C, KDPE-C and Tr-KDPE-C with DDIM sampling on the machine used for the real-world experiments (equipped with an NVIDIA RTX 3080 GPU). DP-C, which in the original implementation generates a single trajectory, requires 77ms, while generating a population of 100 trajectories takes an additional 13ms, and performing KDE on them adds only 3ms. Therefore, DP-C can be executed at a control frequency of $\sim$ 12.99Hz, while KDPE-C at $\sim$ 10.75Hz, meaning that KDPE-C adds a computational overhead of only $\sim$ 2.24Hz. For Tr-KDPE-C, instead, performing KDE requires 30ms, and can be executed at $\sim$ 8.33Hz.

8.6 Conclusion

DP has recently gained popularity as one of the most effective methods to train behavior cloning policies, thanks to its policy parameterization as a DDPM which allows to model the multimodality in the training data. While being effective, however, the sampling process is not constrained in any way. This can lead the model to predict trajectories that take the robot out-of-distribution. We propose to overcome this limitation with KDPE, a strategy to select with KDE the most representative trajectory computed by different DP denoising processes.

We quantitatively benchmarked KDPE against DP on four RoboMimic and three MimicGen simulated tasks, and on three real robot experiments: a tabletop plush picking, a multimodal and multi-step cube sorting, and a long-horizon coffee making task that requires high precision to be completed.

We showed that KDPE achieves better performance in the lower demonstration-quality regime and on tasks that require higher precision, also in presence of visual environment perturbations.

Two interesting directions for future research are using KDPE to guide the denoising process of DP, and applying KDPE to higher-dimensional problems, such as bimanual or dexterous manipulation tasks with anthropomorphic hands. However, such applications could

incur the additional challenge of addressing the curse of dimensionality for KDE, a known challenge for kernel-based methods.

8.7 Limitations

Limiting assumptions KDPE selects certain output trajectories among the ones produced by DP. If this latter does not generate a representative distribution of trajectories, e.g. due to poor training, KDPE cannot provide any performance improvement.

Failure modes One of the strengths of DP is that the stochasticity of the trajectory generation can help the robot recover from out-of-distribution states by generating trajectories that may be associated to smaller probability density. KDPE aims to reduce the probability that the robot ends up in such states. However, detecting when this happens (e.g., from the estimated PDF from KDE) and switching to a different trajectory selection method would be an interesting improvement to KDPE.

Limitations of the results and experiments In this paper we proposed KDPE as a component that can be plugged on top of DP. However, it would be interesting to apply KDPE to other generative models like flow matching (Zhang et al., 2023), or to generalist policies (Black et al., 2025a; Ghosh et al., 2024; Kim et al., 2024).

Supplementary Material

8.8 Analysis Under Visual Environment Perturbations

In the experiments under visual environment perturbation presented in Section 8.4.1, we slightly modify the color of an object in each environment. Specifically, if the object has a texture we compute its average color, we reduce its lightness (third channel under the HSL color scheme) by 10% and set it as the object color. The object modified in the simulated task environments are shown in Figure 8.4.

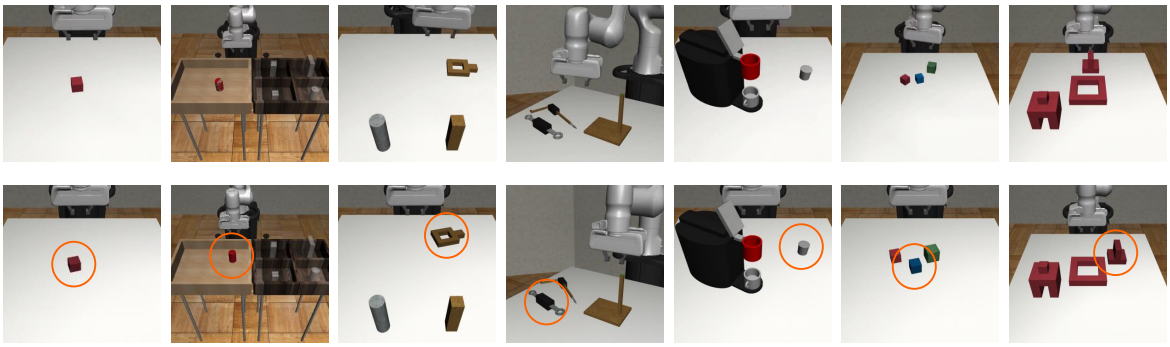


Figure 8.4 First row: unperturbed task environments. Second row: perturbed environments with the objects modified in the color perturbation experiments highlighted in the orange circles.

8.9 Hyperparameters

The hyperparameters used during training and evaluation are reported in Table 8.4.

8.10 DP Standard Deviations

As reported in Section 8.3.2, since the baseline performance (DP-C or DP-T) depends on the random sampling of a trajectory, we repeated its evaluation three times with three different seeds. We report the performance of all the methods on the benchmark experiments with DP standard deviations in Table 8.5, and on the experiments under object color perturbation in Table 8.6.

Hyperparameter	*Value*
Optimizer	Adam
Learning Rate	1×10^{-4}
Betas	$(0.95, 0.999)$
Epsilon	1×10^{-8}
Weight Decay	1×10^{-6}
Noise scheduler	DDPM (simulation) / DDIM (real robot)
Diffusion Inference Steps	100 (simulation) / 10 (real robot)
Total Training Steps	80,000
Population Size (N)	100
Execution Horizon (T)	8
Action Dimensionality (D)	10
Position Bandwidth (σ_{pos})	0.05
Rotation Bandwidth (σ_{rot})	0.25
Gripper Bandwidth (σ_{grip})	1.0

Table 8.4 Hyperparameters used for training and evaluation of KDPE. Symbols in the parentheses correspond to those used in Section 8.3.

8.11 Visualizer

During the development and study of KDPE and comparative baselines, we developed a visualizer using the *rerun* data visualization library (Rerun Development Team, 2024). This tool facilitates the analysis of populations of trajectories, including action positions, orientations, and gripper states. Through the help of the visualizer, we studied how the KDE bandwidths (reported in the last three rows of Table 8.4) influence the action densities, and found suitable values that let KDE capture DP output multimodality. Figure 8.5 shows a snapshot of the visualizer.

We open-source the code of the visualizer, hoping it will assist others in the analysis of generative robotic policies. The code for the visualizer is available on the project page at https://hsp-iit.github.io/KDPE/.

	Lift		Can		*Square*		*ToolHang*
	ph	*mh*	*ph*	*mh*	*ph*	*mh*	*ph*
DP-C	**100 ± 0**	**100 ± 0**	96.7 ± 1.00	94.0 ± 2.00	**93.0 ± 3.00**	83.0 ± 7.00	62.0 ± 2.00
KDPE-OOD-C	**100**	**100**	96.0	84.0	83.0	49.0	7.00
Tr-KDPE-C	**100**	**100**	97.0	94.0	90.0	78.0	68.0
KDPE-C	**100**	**100**	**98.0**	**96.0**	92.0	**86.0**	**76.0**
DP-T	**100 ± 0**	**100 ± 0**	96.0 ± 2.00	90.0 ± 3.00	84.0 ± 4.00	73.0 ± 7.00	61.3 ± 10.0
KDPE-OOD-T	**100**	99.0	95.0	91.0	84.0	65.0	2.00
Tr-KDPE-T	**100**	99.0	93.0	92.0	81.0	81.0	62.0
KDPE-T	**100**	**100**	**97.0**	**92.0**	**88.0**	**83.0**	**66.0**

	Coffee	*Stack*	*Assembly*
DP-C	85.0 ± 6.00	80.7 ± 1.00	54.3 ± 1.00
KDPE-OOD-C	85.0	72.0	40.0
Tr-KDPE-C	76.0	78.0	**61.0**
KDPE-C	**88.0**	**85.0**	**61.0**
DP-T	91.7 ± 6.00	73.7 ± 2.00	**23.3 ± 2.00**
KDPE-OOD-T	**93.0**	61.0	23.0
Tr-KDPE-T	86.0	**74.0**	18.0
KDPE-T	91.0	72.0	23.0

Table 8.5 Benchmark results presented in Table 8.1, reporting also the standard deviation of the success rate (%) over three runs of DP. Top table: RoboMimic (*Lift*, *Can*, *Square*, *ToolHang*). Bottom table: MimicGen (*Coffee*, *Stack*, *Assembly*).

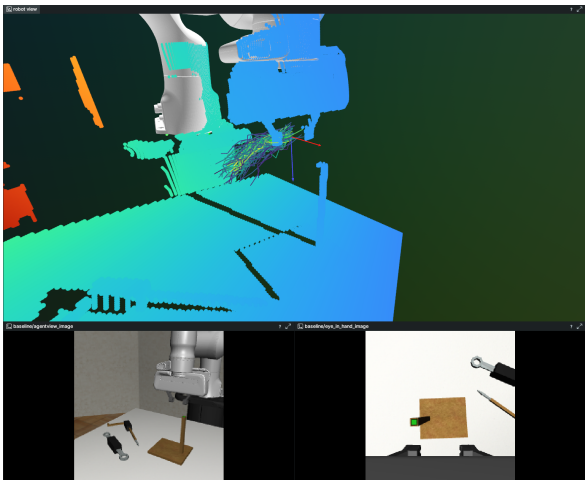


Figure 8.5 The trajectory visualizer being used to analyze trajectories on the RoboMimic *ToolHang* task. The scene in the 3D view (*robot view* window) is represented as a point-cloud, since object meshes are not readily available for real-world environments. The visualizer supports assigning different colormaps to the population of trajectories. The colormap in the picture represents the densities assigned by KDPE to each trajectory.

	Lift		Can		*Square*		*ToolHang*
	ph	*mh*	*ph*	*mh*	*ph*	*mh*	*ph*
DP-C	**100 ± 0**	99.6 ± 0.57	92.3 ± 4.16	**91.3 ± 3.21**	93.7 ± 2.52	79.7 ± 4.16	59.0 ± 1.53
KDPE-C	**100**	**100**	**94.0**	90.0	**94.0**	**90.0**	**69.0**
DP-T	98.7 ± 0.58	**100 ± 0**	89.0 ± 4.58	84.0 ± 3.61	**79.3 ± 1.53**	68.3 ± 2.51	58.0 ± 5.57
KDPE-T	**99.0**	**100**	**90.0**	**85.0**	75.0	**77.0**	**62.0**

	Coffee	*Stack*	*Assembly*
DP-C	92.3 ± 5.03	75.3 ± 0.58	44.7 ± 1.50
KDPE-C	**97.0**	**88.0**	**50.0**
DP-T	**73.3 ± 7.37**	67.7 ± 4.16	15.6 ± 4.16
KDPE-T	72.0	**68.0**	**20.0**

Table 8.6 Results of experiments under object color perturbation presented in Table 8.2, reporting also the standard deviation of the success rate (%) over three runs of DP. Top table: RoboMimic (*Lift*, *Can*, *Square*, *ToolHang*). Bottom table: MimicGen (*Coffee*, *Stack*, *Assembly*).

Part IV

Conclusion

Chapter 9

Conclusion

In this thesis, I identified three key research areas that play an important role in building general robotic manipulation systems: 3D perception, semantic understanding, and motion generation. In each area, I identified the limitations in the existing literature and addressed them with novel methods and benchmarks.

For 3D perception, I developed a shape completion algorithm that could generalize to unseen objects, with the goal of being usable as a perception module in a robotic grasping system. The method outputs an implicit occupancy function and, unlike point-regression approaches, enables sampling reconstructions at arbitrary resolution. A key feature of this representation is that it provides a confidence value over the output space, which can be used to extract a point cloud from the implicit function. The algorithm outperformed the compared state-of-the-art methods on shape completion benchmarks and achieved a higher grasp success rate on a set of previously unseen objects.

At least two future directions follow from this work. First, confidence could be exploited more directly in downstream manipulation: rather than using it only to extract a point cloud, it could condition grasp detection or grasp selection, biasing grasping candidates toward regions where the completion model is more certain. Second, confidence could be used as a signal for active perception or view planning, where the robot moves the sensor to reduce uncertainty before committing to a grasp.

For semantic understanding and personalization, I studied how vision-language models can be adapted to user-specific concepts from only a few images, and how these learned concepts can be composed with a wide range of known contexts. To evaluate this setting, my coauthors and I introduced ConCon-Chi, a dataset designed around two properties that are central to personalization: learning new meanings (modeled with chimeric concepts rather than instances of common categories) and compositionality evaluation (enforced via

a concept-context matrix structure). This design also helps reduce the risk of evaluating methods that succeed mainly by exploiting context or concept bias. By benchmarking state-of-the-art approaches for personalized text-to-image retrieval and generation on ConCon-Chi, I showed that current methods still struggle to reliably compose learned concepts with diverse contexts, especially when dealing with novel concepts. The availability of real images for each concept-context prompt also enables a more faithful evaluation of personalized generation than comparing only against training examples, and it exposes gaps in realism and diversity that are not always evident from standard prompt-only evaluation protocols.

I hope that ConCon-Chi will impact future work by providing a fair benchmark for concept learning and image understanding.

For motion generation, I addressed a limitation of implicit behavior cloning policies: the sampling procedure at inference time can occasionally produce trajectories that are outliers with respect to the dominant modes in the demonstrations, which may push the robot out-of-distribution. I proposed KDPE, an inference-time strategy that samples multiple candidate trajectories and selects representative ones using kernel density estimation. The density estimator uses a manifold-aware kernel defined over end-effector pose and gripper state, enabling a measure of closeness between actions.

Across simulated benchmarks and real-robot experiments, KDPE on average improved success rates over the baseline with particularly strong performance in lower demonstration-quality settings and in tasks requiring higher precision.

These findings open several directions for future research. First, extending KDE-based selection to higher-dimensional action spaces (e.g., bimanual or dexterous manipulation) may suffer from the curse of dimensionality, motivating research on new techniques to characterize the actions. Second, this work highlights test-time computing as a way to improve robotic policies' performance: generating multiple action hypotheses and re-ranking them with task-relevant criteria could be applied beyond diffusion policies.

Across these contributions, a recurring theme is the use of test-time procedures to obtain the final output of a learned model. In shape completion, the predicted confidence guides how a point cloud is extracted from the implicit representation; in robot learning, KDE provides density estimates that determine which trajectories are selected.

Finally, another possible connection is with vision-language-action (VLA) models, i.e., policies that map multimodal inputs (images and language) to robot actions. Personalization methods that map user-provided images to new textual tokens (Section 7.3) could be used to specify tasks involving unseen objects via compositional instructions (e.g., "pick

[pers_concept_A] and place it on [pers_concept_B]") even when the VLA was not trained on those concepts.

References

Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 689–723. PMLR, 2025. 85

Dzmitry Bahdanau, Tom Bosc, Stanisław Jastrzębski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. Learning to Compute Word Embeddings On the Fly. *arXiv preprint arXiv:1706.00286*, 2017. 49

Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-Shot Composed Image Retrieval with Textual Inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15338–15347, October 2023. 3, 48, 50, 55, 56, 57, 58, 60, 76

Kuldeep R Barad, Antoine Richard, Jan Dentler, Miguel Olivares-Mendez, and Carol Martinez. Object-centric reconstruction and tracking of dynamic unknown objects using 3d gaussian splatting. In *2024 International Conference on Space Robotics (iSpaRo)*, pages 202–209. IEEE, 2024. 3

Timothy D. Barfoot and Paul T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014. doi: 10.1109/TRO.2014.2298059. 25, 83, 87

Steven M. Beitzel, Eric C. Jensen, and Ophir Frieder. Map. In LING LIU and M. TAMER ÖZSU, editors, *Encyclopedia of Database Systems*, pages 1691–1692. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_492. URL https://doi.org/10.1007/978-0-387-39940-9_492. 56

Stefano Berti, Andrea Rosasco, Michele Colledanchise, and Lorenzo Natale. One-shot open-set skeleton-based action recognition. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 765–772, 2022. doi: 10.1109/Humanoids53995.2022.10000115. 26

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang,

Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025a. URL https://openreview.net/forum?id=vlhoswksBO. 84, 96

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010. 84

Jeannette Bohg, Matthew Johnson-Roberson, Beatriz León, Javier Felip, Xavi Gratal, Niklas Bergström, Danica Kragic, and Antonio Morales. Mind the gap-robotic grasping under incomplete observation. In *2011 IEEE international conference on robotics and automation*, pages 686–693. IEEE, 2011. 33

Richard Bormann, Bruno Ferreira de Brito, Jochen Lindermayr, Marco Omainska, and Mayank Patel. Towards automated order picking robots for warehouses and retail. In *International Conference on Computer Vision Systems*, pages 185–198. Springer, 2019. 32

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Manon Devin, Alex X. Lee, Maria Bauza Villalonga, Todor Davchev, Yuxiang Zhou, Agrim Gupta, Akhil Raju, Antoine Laurens, Claudio Fantacci, Valentin Dalibard, Martina Zambelli, Murilo Fernandes Martins, Rugile Pevceviciute, Michiel Blokzijl, Misha Denil, Nathan Batchelor, Thomas Lampe, Emilio Parisotto, Konrad Zolna, Scott Reed, Sergio Gómez Colmenarejo, Jonathan Scholz, Abbas Abdolmaleki, Oliver Groth, Jean-Baptiste Regli, Oleg Sushkov, Thomas Rothörl, Jose Enrique Chen, Yusuf Aytar, David Barker, Joy Ortiz, Martin Riedmiller, Jost Tobias Springenberg, Raia Hadsell, Francesco Nori, and Nicolas Heess. Robocat: A self-improving generalist agent for robotic manipulation. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=vsCpILiWHu. 84

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems*, 2023. ISBN 978-0-9923747-9-2. URL http://dblp.uni-trier.de/db/conf/rss/rss2023.html. 84

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. 85

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 13, 39

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 8

Vinod Kumar Chauhan, Sukhdeep Singh, and Anuj Sharma. Hcr-net: A deep learning based script independent handwritten character recognition network. *Multimedia Tools and Applications*, 83(32):78433–78467, 2024. 10

Wentao Chen, Chenyang Si, Zhang Zhang, Liang Wang, Zilei Wang, and Tieniu Tan. Semantic Prompt for Few-Shot Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23581–23591, June 2023. 50

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 4, 22, 83, 84, 85, 89

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 8, 32

Niv Cohen, Rinon Gal, Eli A. Meirom, Gal Chechik, and Yuval Atzmon. "This Is My Unicorn, Fluffy": Personalizing Frozen Vision-Language Representations". In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 558–577, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20044-1. 3, 47, 49, 57

D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. doi: 10.1109/34.1000236. 83

Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017. 8

Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024. 85

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602 (7897):414–419, 2022. 21

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 9, 34

Kester Duncan, Sudeep Sarkar, Redwan Alqasemi, and Rajiv Dubey. Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 4238–4243, 2013. doi: 10.1109/ICRA.2013.6631176. 7

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996. 83

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 158–168. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/florence22a.html. 4, 22

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=NAQvF08TcyG. 3, 17, 47, 50, 63, 64, 65

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based Domain Tuning for Fast Personalization of Text-to-Image Models. *ACM Trans. Graph.*, 42(4), jul 2023b. ISSN 0730-0301. doi: 10.1145/3592133. URL https://doi.org/10.1145/3592133. 18, 50

Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5337–5345, 2019. 48, 49

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.090. 84, 85, 96

Igor Gilitschenski, Gerhard Kurz, Simon J. Julier, and Uwe D. Hanebeck. A new probability distribution for simultaneous representation of uncertain position and orientation. In *17th International Conference on Information Fusion (FUSION)*, pages 1–7, 2014. 83

Corey Goldfeder, Matei Ciocarlie, Hao Dang, and Peter K Allen. The columbia grasp database. In *2009 IEEE international conference on robotics and automation*, pages 1710–1716. IEEE, 2009. 13, 39

Jia Guo, Xuanxia Yao, Mengyu Shen, Jiafei Wang, and Wanyou Liao. A deep learning network for point cloud of medicine structure. In *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pages 683–687, 2018. doi: 10.1109/ITME.2018.00157. 7

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx. 10, 11, 30, 33

Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023. 85

Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022. doi: 10.1109/MRA.2021.3138382. 4, 84

Jiyeon Han, Hwanil Choi, Yunjey Choi, Junho Kim, Jung-Woo Ha, and Jaesik Choi. Rarity Score: A New Metric to Evaluate the Uncommonness of Synthesized Images. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JTGimap_-F. 50, 66

Aurélie Herbelot and Marco Baroni. High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1030. URL https://aclanthology.org/D17-1030. 49

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016. doi: 10.1162/tacl_a_00080. URL https://aclanthology.org/Q16-1002. 49

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 16, 23, 83

Rob J. Hyndman, David M. Bashtannyk, and Gary K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4):315–336, 1996. ISSN 1061-8600, 1537-2715. doi: 10.1080/10618600.1996.10474715. URL http://www.tandfonline.com/doi/abs/10.1080/10618600.1996.10474715. 89

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016. 33

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Xinyu Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Khuong Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024. URL https://openreview.net/forum?id=Ml2pTYLNLi. 84

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6. 84, 96

Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 34

Bruno Korbar and Andrew Zisserman. Personalised CLIP or: how to find your vacation videos. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL https://bmvc2022.mpi-inf.mpg.de/0639.pdf. 49

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh. 54

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-Concept Customization of Text-to-Image Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1941, June 2023. 18, 19, 50

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved Precision and Recall Metric for Assessing Generative Models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0234c510bc6d908b28c70ff313743079-Paper.pdf. 50

Angeliki Lazaridou, Marco Marelli, and Marco Baroni. Multimodal Word Meaning Induction From Minimal Exposure to Natural Text. *Cognitive Science*, 41(S4):677–705, 2017. doi: https://doi.org/10.1111/cogs.12481. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12481. 19, 48, 49

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/li22n.html. 47

Shidi Li, Miaomiao Liu, and Christian Walder. Editvae: Unsupervised parts-aware controllable 3d point cloud shape generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1386–1394, 2022b. 7

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Arsh Koneru, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Reflect-dit: Inference-time scaling for text-to-image diffusion transformers via in-context reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15657–15668, 2025. 85, 86

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 50, 57

Gidi Littwin and Lior Wolf. Deep meta functionals for shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1824–1833, 2019. 10, 11, 33, 37

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021a. 34

Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image Retrieval on Real-life Images with Pre-trained Vision-and-Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2125–2134, 2021b. 50, 55

Zijun Long, Xuri Ge, Richard McCreadie, and Joemon M Jose. Cfir: Fast and effective long-text to image retrieval for large corpora. In *Proceedings of the 47th international acm sigir conference on research and development in information retrieval*, pages 2188–2198, 2024. 15

Jens Lundell, Francesco Verdoja, and Ville Kyrki. Robust grasp planning over uncertain shape completions. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1526–1532. IEEE, 2019. 13, 33, 39, 41

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Scaling inference time compute for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2523–2534, 2025. 85, 86

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. CREPE: Can Vision-Language Foundation Models Reason Compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10910–10921, June 2023. 51

Jeffrey Mahler, Florian T Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1957–1964. IEEE, 2016. 33

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *5th Annual Conference on Robot Learning*, 2021. 4, 25, 89

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pages 1678–1690. PMLR, 2022. 84

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023. 25, 84, 89

David Q Mayne and Hannah Michalska. Receding horizon control of nonlinear systems. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 464–465. IEEE, 1988. 83

Sachit Menon and Carl Vondrick. Visual Classification via Description from Large Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jlAjNL8z5cs. 47

Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*, 3(1):8, 2006. doi: 10.5772/5761. URL https://doi.org/10.5772/5761. 26

Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004. 13

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023. 3

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable Fidelity and Diversity Metrics for Generative Models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7176–7185. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/naeem20a.html. 50, 66, 78

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=6FGlpzC9Po. 85

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 84

Weihua Ou, Yingjie Chen, Linqing Liang, Jianping Gou, Jiahao Xiong, Jiacheng Zhang, Lingge Lai, and Lei Zhang. Cross-modal retrieval of chest x-ray images and diagnostic reports based on report entity graph and dual attention. *Multimedia Systems*, 31(1):58, 2025. 15

Sarah Parisot, Yongxin Yang, and Steven McDonagh. Learning To Name Classes for Vision and Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23477–23486, June 2023. 49

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 9

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018. 9

Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. 23, 87

Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3d scan completion. In *Proceedings of the Third Eurographics Symposium on Geometry Processing*, SGP '05, page 23–es, Goslar, DEU, 2005. Eurographics Association. ISBN 390567324X. 8

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 8, 32

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html. 3, 15, 16, 47, 49, 56

Rerun Development Team. Rerun: A visualization sdk for multimodal data, 2024. URL https://www.rerun.io. Available from https://www.rerun.io/ and https://github.com/rerun-io/rerun. 98

Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-Conditioned Imitation Learning using Score-based Diffusion Policies. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.028. 85

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 (CVPR)*, pages 10684–10695, June 2022. 16, 18, 23, 47, 50, 64

Andrea Rosasco, Stefano Berti, Fabrizio Bottarel, Michele Colledanchise, and Lorenzo Natale. Towards confidence-guided shape completion for robotic applications. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 580–586, 2022. doi: 10.1109/Humanoids53995.2022.10000218. 7

Andrea Rosasco, Stefano Berti, Giulia Pasquale, Damiano Malafronte, Shogo Sato, Hiroyuki Segawa, Tetsugo Inada, and Lorenzo Natale. Concon-chi: Concept-context chimera benchmark for personalized vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22239–22248, June 2024. 5, 14

Andrea Rosasco, Federico Ceola, Giulia Pasquale, and Lorenzo Natale. Kdpe: A kernel density estimation strategy for diffusion policy trajectory selection. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pages 1210–1224. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/rosasco25a.html. 21

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 85

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510, 2023. 3, 18, 19, 50, 51, 52, 63, 64, 66

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 58

Elena Sofia Ruzzetti, Leonardo Ranaldi, Michele Mastromattei, Francesca Fallucchi, Noemi Scarpato, and Fabio Massimo Zanzotto. Lacking the Embedding of a Word? Look it up into a Traditional Dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2651–2662, Dublin, Ireland, May 2022. Association for Computational

Linguistics. doi: 10.18653/v1/2022.findings-acl.208. URL https://aclanthology.org/2022.findings-acl.208. 49

Som Sagar, Jiafei Duan, Sreevishakh Vasudevan, Yifan Zhou, Heni Ben Amor, Dieter Fox, and Ransalu Senanayake. From mystery to mastery: Failure diagnosis for improving manipulation policies. In *Second Workshop on Out-of-Distribution Generalization in Robotics at RSS 2025*, 2025. URL https://openreview.net/forum?id=3d05tYQHD9. 85

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 16

Kuniaki Saito, Kihyuk Sohn, Xiang Zhang, Chun-Liang Li, Chen-Yu Lee, Kate Saenko, and Tomas Pfister. Pic2Word: Mapping pictures to words for zero-shot composed image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19305–19314, 2023. 3, 17, 19, 48, 50, 56, 57, 58

Ruwen Schnabel, Patrick Degener, and Reinhard Klein. Completion and reconstruction with primitive shapes. *Computer Graphics Forum*, 28(2):503–512, 2009. 7, 33

Simon Schug, Seijin Kobayashi, Yassir Akram, Joao Sacramento, and Razvan Pascanu. Attention as a hypernetwork. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Representation Learning*, volume 2025, pages 68744–68770, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/abfa542f546df3c6c35695ec8d5bf4b9-Paper-Conference.pdf. 10

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=agTr-vRQsa. 4, 22

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238. 58

David Silver, Aja Huang, Chris J. Maddison, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. doi: 10.1038/nature16961. 21

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n. 85

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP. 23, 92

Otilia Stretcu, Edward Vendrow, Kenji Hata, Krishnamurthy Viswanathan, Vittorio Ferrari, Sasan Tavakkol, Wenlei Zhou, Aditya Avinash, Emming Luo, Neil Gordon Alldrin, MohammadHossein Bateni, Gabriel Berger, Andrew Bunner, Chun-Ta Lu, Javier Rey, Giulia DeSalvo, Ranjay Krishna, and Ariel Fuxman. Agile Modeling: From Concept to Classifier in Minutes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22323–22334, October 2023. 47

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A Corpus for Reasoning about Natural Language Grounded in Photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1644. URL https://aclanthology.org/P19-1644. 50

Minhyuk Sung, Vladimir G. Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Trans. Graph.*, 34(6), November 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818094. URL https://doi.org/10.1145/2816795.2818094. 8

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. 84

Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017a. 10

Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017b. 35

Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-Locked Rank One Editing for Text-to-Image Personalization. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi: 10.1145/3588432.3591506. URL https://doi.org/10.1145/3588432.3591506. 18, 50, 51, 52, 54, 64, 65, 66

Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning*, pages 306–316. PMLR, 2018. 3, 32

Mark Van der Merwe, Qingkai Lu, Balakumar Sundaralingam, Martin Matak, and Tucker Hermans. Learning continuous 3d reconstructions for geometrically aware grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11516–11522. IEEE, 2020. 33

Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017. 7, 13, 33, 38, 39, 41

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 9, 33

Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing Text and Image for Image Retrieval - an Empirical Odyssey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6439–6448, June 2019. 48

Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgwNerKvB. 10

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 64

Xiaodan Wang, Chengyu Wang, Lei Li, Zhixu Li, Ben Chen, Linbo Jin, Jun Huang, Yanghua Xiao, and Ming Gao. FashionKLIP: Enhancing E-commerce image-text retrieval with fashion multi-modal conceptual knowledge graph. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 149–158, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.16. URL https://aclanthology.org/2023.acl-industry.16/. 15

Xintong Wang, Xiaoyu Li, Xingshan Li, and Chris Biemann. Probing large language models from a human behavioral perspective. In *Proceedings of the Workshop: Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning (NeusymBridge)@ LREC-COLING-2024*, pages 1–7, 2024. 4

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 8, 9, 37

Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023. 3

Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. Fashion IQ: A New Dataset Towards Retrieving Images by Natural Language Feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11307–11317, 2021. 50

Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–662, 2018. 8, 32

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=j7DZWSc8qu. 85

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 1912–1920. IEEE Computer Society, oct 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298801. 8

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018. 3

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 68578–68598. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/xie25b.html. 85, 86

Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-Sequence Video Object Segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 603–619, Cham, 2018. Springer International Publishing. 49

Daniel Yang, Tarik Tosun, Benjamin Eisner, Volkan Isler, and Daniel Lee. Robotic grasping through combined image-based grasp proposal and 3d reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6350–6356. IEEE, 2021. 33

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018. 9, 35

Chun-Hsiao Yeh, Bryan Russell, Josef Sivic, Fabian Caba Heilbron, and Simon Jenni. Meta-Personalizing Vision-Language Models To Find Named Instances in Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19123–19132, June 2023. 49

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021. 9, 11, 34

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 35

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information*

Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf. 57

Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkgW0oA9FX. 33

Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, and Joseph J Lim. Bootstrap your own skills: Learning to solve new tasks with large language model guidance. In *7th Annual Conference on Robot Learning*, 2023. 85, 96

Xiaoyu Zhang, Matthew Chang, Pranav Kumar, and Saurabh Gupta. Diffusion Meets DAgger: Supercharging Eye-in-hand Imitation Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.048. 85

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746, 2019. doi: 10.1109/CVPR.2019.00589. 87

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html. 84