Deep Learning for Dexterous Robot Grasping

Hrishit Leen, Kunal Aneja*, Chetan Reddy, Priyadarshini Tamilselvan, Nhi Nguyen, Sri Siddarth Chakaravarthy, Jeremy Collins, Miroslav Bogdanovic, Animesh Garg*

Georgia Institute of Technology, Atlanta, GA, USA
Emails: {hleen3, kunala, creddy32, ptamilselvan3, nnguyen349, sp313, jer, animesh.garg}@gatech.edu

Keywords: *Dexterous Grasping, Differentiable Simulation, Multi-Fingered Robot Hands, Task-Oriented Manipulation, Sim-to-Real Transfer*

This paper presents a comprehensive survey of deep learning approaches for dexterous robotic grasping, emphasizing recent progress enabled by multi-modal models and data-driven techniques. These developments have enabled the generation and execution of stable, context-aware grasps that can be conditioned on natural language, generalize across robot embodiments, and perform effectively in real-world settings. We organize our survey into three parts: (1) **Datasets**, the foundation for data-driven approaches, covering large-scale efforts that support learning-based grasping; (2) **Grasp Synthesis**, including diverse representation methods, generative modeling, and optimization-based techniques; and (3) **Grasp Execution**, encompassing reinforcement learning, imitation learning, heuristic control, and hybrid frameworks that translate grasps into executable actions. We also examine existing benchmarks and metrics for evaluating grasp plausibility, stability, and task alignment. We also identify persistent challenges that bottleneck progress and discuss promising future directions to guide researchers toward building more general-purpose, robust dexterous manipulation systems. More details at DexRobotGraspSurvey.github.io

1 Introduction

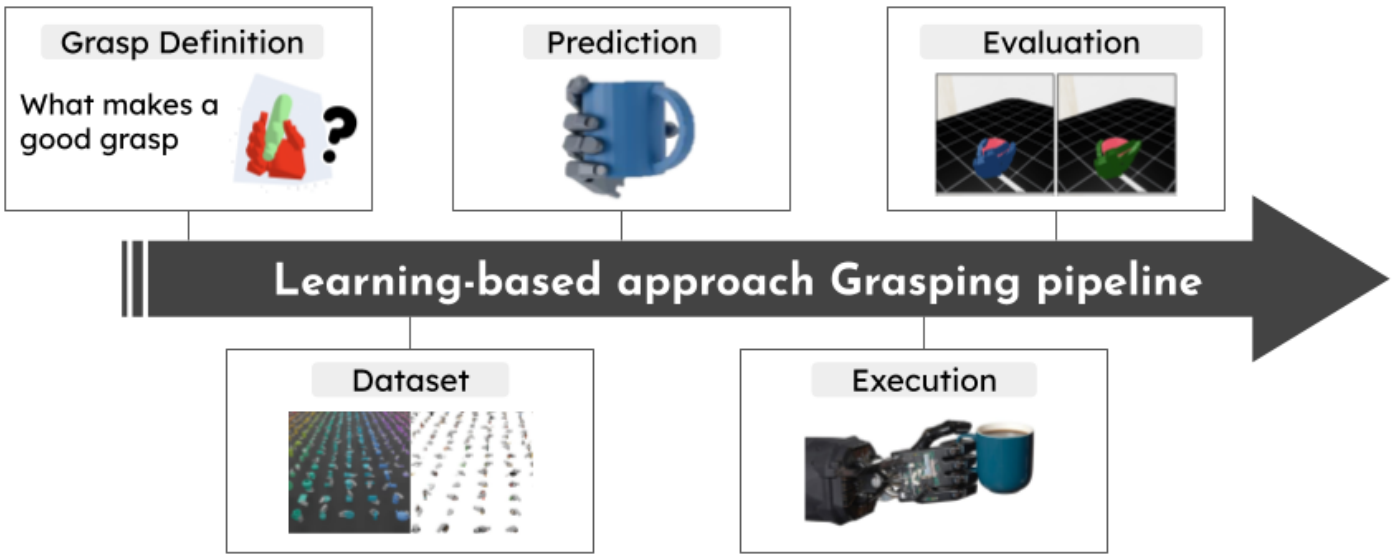


Figure 1: Diagram depicting the learning-based grasping pipeline and the corresponding sections of this survey paper

The leap from rigid two-finger grippers to multi-fingered robot hands is a key step to generalizable robotic manipulation. Despite a growth in dexterous hands, true dexterity remains a tantalizing mirage outside controlled labs. Why is that, and what have we learned in the past five years that finally moves us closer? Imagine a multi-fingered hand peeling a banana for you or a service bot twisting a screwdriver—capabilities still out of reach for most state-of-the-art grippers. Closing this gap demands not only smarter hardware but a fundamentally new way of collecting data, learning grasps, and executing them reliably in the wild.

This survey distills the rapidly growing body of *learning-based* dexterous grasping work into a coherent narrative. Our goal is two-fold: (i) provide newcomers a guided map through otherwise fragmented

literature and (ii) give practitioners a reference checklist of datasets, synthesis models, execution policies, and evaluation tools for the grasps.

The idea of formally *measuring* a good grasp is almost forty years old: Ferrari and Canny's ϵ-quality (1992) quantified how much external wrench a grasp can resist. Throughout the 1990s and early 2000s, researchers relied on these analytical metrics plus brute-force search in small pose grids—an approach that scaled poorly once anthropomorphic hands with 20+ degrees of freedom entered the scene.

However, the design and control of dexterous hands present significant challenges. These include high-dimensional action spaces, intricate contact dynamics, and the difficulty of generalizing across diverse object categories and tasks. Traditional methods based on analytical grasp metrics or precomputed databases often struggle with scalability and robustness in the face of this complexity.

To overcome these limitations, recent advances in deep learning, differentiable simulation, and large-scale dataset curation have enabled the development of data-driven approaches for dexterous grasp synthesis and execution. These learning-based techniques not only improve robustness and generalization but also open the door to multimodal conditioning—allowing grasps to be specified using natural language, visual cues, affordances, or task intent. This capability is particularly useful for intuitive human-robot interaction and task-aware manipulation.

Today the field straddles two worlds: *scalable simulation* that offers unlimited data yet imperfect realism, and limited but irreplaceable *physical trials* that expose unmodeled friction, latency, and wear. Bridging these worlds forms the central arc of the pages that follow.

Despite this progress, several open challenges remain. Chief among them is the sim-to-real transfer gap, where models trained in simulation often fail to perform reliably on real-world hardware due to unmodeled physical nuances. Another key issue is the lack of standardized benchmarks, which hampers fair comparison across different robot embodiments and algorithmic paradigms.

1.1 Executive Overview

This survey focuses on the role of learning-based approaches in robotic grasping and is organized around five pillars: (1) defining what makes a good grasp, (2) analyzing grasp datasets, (3) methods to predict grasps, (4) executing grasps with learned policies, and (5) evaluating grasp performance in simulation.

Datasets 3 The central idea is that robust and versatile grasping capabilities require large and diverse datasets, which are primarily acquired through two main avenues: simulation and real-world human hand-object interaction.

Simulation-based datasets are further divided into two categories. Sampling-based methods (3.1.1), generate a multitude of potential grasps and then filter for quality using analytical metrics. While effective, this can be inefficient. More recently, differentiable simulation (3.1.2) uses gradient-based optimization to more efficiently generate high-quality grasps. Originally, datasets derived from human interaction offer the benefit of demonstrating natural, functional, and context-aware grasping behaviors, even though they may be smaller in scale than simulated datasets. These are often captured using motion capture systems or egocentric cameras. There is also an emerging trend in conditional grasp generation (3.3), where datasets are augmented with contextual information, such as object affordances or natural language descriptions, to enable robots to generate grasps that are not just stable but also appropriate for a specific task or instruction.

Learning to Predict Grasps 4 Generating stable and functional grasps for multi-fingered robot hands is a complex, high-dimensional problem that researchers are tackling by making critical choices along three main axes: how the grasp is represented (4.1), the type of deep learning architecture used (4.2), and

the methods for refining the final output (4.3). We analyze these components to explain how different strategies are combined to create effective grasp generation models.

The initial and most critical choice is the grasp representation, which dictates how the model "thinks" about the problem. This can range from predicting contact maps on the object's surface to directly outputting hand parameters (like joint angles), or using more sophisticated geometrically aware representations that embed the hand and object shapes into the learning process. The choice of the machine learning backbone architecture—such as versatile transformers for handling diverse inputs, generative models like diffusion for creating varied and high-quality outputs, or Autoencoders for learning a compressed representation of grasps—is also crucial. Finally, regardless of the initial generation method, nearly all approaches require an optimization and refinement step to ensure the resulting grasps are physically plausible, stable, and free of collisions, often using physics-based metrics as loss functions or filtering candidates through simulation.

Grasp Execution 5

The final stage of the robotic grasping pipeline where a manipulator must securely pick, hold, and manipulate an object. We detail the evolution from rigid, predefined controllers to sophisticated, adaptive policies driven by machine learning (5.4.2). These modern approaches are essential for enabling dexterous hands to perform complex, task-specific actions in unstructured real-world environments, moving far beyond simple pick-and-place operations.

We break down modern strategies into three main categories. The first is Reinforcement Learning (RL), where a robot learns optimal grasping behaviors through trial-and-error, guided by a reward function that encourages stability and task success (5.4.3). The second is Imitation Learning (IL), which circumvents the difficult exploration problem of RL by training policies to mimic expert demonstrations, though it faces challenges like translating human motion to a robot's body (5.5). The most powerful and common trend, however, is the use of hybrid approaches that combine the strengths of both. These methods often use expert data from IL to "warm-start" or guide the RL process, dramatically improving sample efficiency and enabling the robot to learn robust, nuanced, and functional grasps that are both physically plausible and task-appropriate.

Together, these developments represent a shift from manually engineered grasp strategies to context-aware, data-driven dexterous manipulation—paving the way toward general-purpose robotic hands that can operate in diverse, real-world environments.

2 What Makes a Good Grasp? Metrics and Evaluation

The grasping space has spawned many works, therefore, it becomes paramount to have widely accepted ways of comparing the performance and efficacy of the new models introduced. Although there is much diversity in the goals and intentions of these papers, the space has adopted commonly used metrics to rank the quality and alignment of grasps produced, allowing for some grounding to compare different grasping works. It's worth noting, as seen in Section 3.1.3, many of these metrics also serve as losses or feedback mechanisms in the optimization and grasp refinement modules for lots of grasp models if they're able to be efficiently computed during the optimization loop. We also mention briefly how these metrics manifest for works mainly tackling grasp synthesis for human/MANO hand models.

2.1 Stability and Plausibility

Although the other metrics discussed might not be necessary or even a valid means of comparison for some grasping works, the physical plausibility and stability of a grasp is a key metric to include if the work has

Category	Metric	Description	Objective
Stability & Plausibility	Simulation Success Rate	Percentage of grasps that remain stable under physics simulation (e.g., with gravity).	↑
	Q1 / Epsilon Quality	Measures the grasp's resistance to external wrenches (forces and torques). A formal measure of force closure.	↑
	Penetration Volume/ Distance	The volume or max distance of intersection between the hand and object mesh. Measures physical plausibility.	↓
Dataset Accuracy	Chamfer Distance (CD)	Measures the distance between the generated grasp's point cloud and the ground truth.	↓
	MPJPE / MRRPE	Mean Per-Joint/Root Position Error. Used for human/MANO hand models to compare against ground truth poses.	↓
Task Alignment / Intention	Fréchet Inception Distance (FID)	Measures the similarity between the distribution of generated grasps and a ground truth set, often on rendered images.	↓
	LLM/VLM Score	Using a VLM (e.g., GPT-4o) to score the semantic consistency between a text prompt and the rendered grasp image.	↑
Diversity	Std. Dev. of Joint Angles	Measures the variation in generated grasps for the same input, indicating if the model produces diverse solutions.	↑

Table 1: Key evaluation metrics for dexterous grasping benchmarks. *For each metric, ↓ represents that a lower quantity is better and ↑ otherwise*

any intention of deploying their model in the real world (which papers tackling robotic dexterous grasping usually do).

Dataset Accuracy:

- **Chamfer Distance (CD)** (↓): Datasets for robotic grasp datasets often include the point-cloud/mesh of the ground truth grasp. The chamfer distance finds the similarity between two sets of points. Given point clouds A and B, it's defined as the sum of distances for each point in A to it's nearest neighbor in B plus the sum of distances for each point in B to it's nearest neighbor in A.
  Formally this is defined as $\mathcal{A} = \{x_i\}_{i=1}^{n}$ and $\mathcal{B} = \{y_j\}_{j=1}^{m}$ be two point clouds in $\mathbb{R}^3$. The Chamfer Distance between $\mathcal{A}$ and $\mathcal{B}$ is defined as

$$\mathrm{CD}(\mathcal{X}, \mathcal{Y}) = \frac{1}{n} \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|_2^2 \; + \; \frac{1}{m} \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|y - x\|_2^2.$$

  Some variants use the L2 norm (without squaring) inside the sums. Intuitively, each point in one cloud is matched to its closest point in the other cloud, and the average squared distance is reported. In grasping, A and B often represent sets of hand or object surface points for predicted vs. ground-truth grasps.
- **Contact Distance** (↓): If contact maps are available in the dataset, it's also useful to characterize the difference between the contact made by the predicted grasp and that from the ground truth contact map. A simple way is to take the L2 norm between the contact maps, also called "Contact Distance" (↓).
  Let $\Omega = (\Omega_1, \ldots, \Omega_N)$ and $\hat{\Omega} = (\hat{\Omega}_1, \ldots, \hat{\Omega}_N)$ be discrete contact maps in $\mathbb{R}^N$ for the ground-truth and predicted grasps, respectively, where each $\Omega_i \in [0, 1]$ measures the contact likelihood at point i. The Contact Distance is defined as

$$\mathrm{ContactDist} = \|\hat{\Omega} - \Omega\|_2^2 = \sum_{i=1}^{N} \big(\hat{\Omega}_i - \Omega_i\big)^2.$$

This scalar quantifies contact consistency: zero means perfect match. In practice, one forms the object contact map by marking object surface points near any hand mesh vertex (e.g. within a threshold) and similarly for the prediction. Minimizing this L2 contact distance encourages the predicted grasp to make contact at the same object regions as the reference

- **Contact Ratio** (↑): Papers that focus on stable grasps that aren't goal or task directed include a "Contact Ratio" (↑) metric that measures the percentage of the object point-cloud that is in contact with the hand. This characterizes the trend that a more stable grasp would usually have more hand-object contact. However, this is a misleading metric for function and task oriented grasps, as the conditioning could require the grasp to make little contact with the object.
  In generation benchmarks, it often denotes the fraction of grasps that achieve any contact. Formally, over a set of grasps G,

$$\text{ContactRatio} = \frac{\big|\{\, g \in G : g \text{ has at least one contact point}\}\big|}{|G|}.$$

- **Mean Per-Joint Position Error** (↓) and **Mean Relative-Root Position Error** (↓): Datasets that are only for a human hand mesh like MANO also include the parameters of the hand model for each grasp (which are not usually found in robot datasets including varying embodiments). When the ground truth parameters of the hand model, it would be more accurate to characterize the accuracy in this space, taking the L2 distance between joint angles and global rotation and translation parameters, named "Mean Per-Joint Position Error" (MPJPE) (↓) and "Mean Relative-Root Position Error" (MRRPE) (↓) respectively.
  Let a dexterous hand have K joints, with ground-truth positions $J^k \in \mathbb{R}^3$ and predicted positions $\hat{J}^k \in \mathbb{R}^3$ for $k = 1, \ldots, K$. Denote the root (e.g. wrist) index by r. MPJPE is the average Euclidean error after aligning both poses at the root:

$$\text{MPJPE} = \frac{1}{K}\sum_{k=1}^{K}\big\|(\hat{J}^k - \hat{J}^r) \;-\; (J^k - J^r)\big\|_2.$$

  For two root joints (e.g. left/right) with ground-truth positions $J^r_{(1)}, J^r_{(2)} \in \mathbb{R}^3$ and predictions $\hat{J}^r_{(1)}, \hat{J}^r_{(2)}$, MRRPE measures the error in their relative displacement:

$$\text{MRRPE} = \big\|\big(\hat{J}^r_{(1)} - \hat{J}^r_{(2)}\big) \;-\; \big(J^r_{(1)} - J^r_{(2)}\big)\big\|_2.$$

Physics Simulation Success:

- **Simulation Displacement** (↓): A common method is to instantiate the grasp in simulation per object in the dataset (eg: ISAAC Sim, Mujoco, RFUniverse, PyBullet, etc.) and enable physics and/or other forces (eg: some papers enable gravity in 6 different directions) and measure the success rate (↑) (whether the object is displaced or not) or the displacement and report the average displacement after a number of trials, named "Simulation Displacement" (SD) (↓).
  Typically one measures the object's center-of-mass shift Δp (or its magnitude) after simulating the grasp and release. For example, many benchmarks use the average final COM displacement:

$$SD = \|p_{\text{final}} - p_{\text{initial}}\|_2,$$

- **Q1** (↑): A formal way of measuring the effort needed to dislodge an object from a grasp is to report the mean of the norm of the minimum wrench force needed to dislodge the object (measured over all possible directions) [154]. This wouldn't be appropriate for models that also try to align a grasp with conditioning, as some conditioning could make the grasp weak to wrenches in a certain direction.
  Given a grasp on a rigid object, one can compute the Grasp Wrench Space (GWS) – the set $W \subset \mathbb{R}^6$ of all wrenches (forces/torques) the grasp can apply. If the grasp achieves force-closure

(can resist any external wrench), the origin lies inside W. Q1 is the radius of the largest ball centered at the origin contained in W (arxiv.org). Equivalently, it is the norm of the smallest external wrench that can "break" the grasp. Mathematically:

$$Q_1 = \max_r \{\, r : \{\, w \in \mathbb{R}^6 : \|w\| \leq r\} \subset W\}.$$

This can be computed by solving a convex optimization: for each contact, consider friction cones and forces, assemble the convex hull of resulting wrenches, then find the inscribed ball radius.

Mesh Penetration:

- **Penetration Distance (PD)** (↓): Penetration Distance quantifies the deepest inter-penetration between the hand mesh H and object mesh O. It is the magnitude of the smallest translation required to separate the two meshes. Using a point-wise penetration depth function $d(x)$ (e.g. from the GJK algorithm),

$$\mathrm{PD}(H, O) = \max_{x \in H \cap O} d(x),$$

where $d(x)$ is how far point $x \in H$ lies inside O. In practice, collision libraries return the maximum triangle-triangle penetration depth; a value of 0 indicates no inter-penetration, while larger PD values signal deeper (physically impossible) mesh penetration.

- **Inter-Penetration Volume** (↓): The Inter-penetration Volume measures the 3-D volume of overlap between the hand and object meshes. Let $\mathrm{Vol}(A)$ denote the Lebesgue volume of region $A \subset \mathbb{R}^3$. Then

$$V_{\text{int}} = \mathrm{Vol}(H \cap O).$$

It is commonly estimated by voxelising both meshes on a fine grid (e.g. $1\,\mathrm{mm}^3$ voxels) and counting overlapping voxels, or via analytic mesh-intersection routines. Ideally $V_{\text{int}} = 0$; larger values indicate more extensive mesh intersections. Together with PD, this provides a fuller picture of grasp physical plausibility.

2.2 Intention

Grasp synthesis models that condition the grasp on a prompt or task of some kind should have a method of measuring how well the grasp aligns with the intention of the conditioning. Empirically, authors can gain qualitative feedback from a group of volunteers that score how well the generated grasp correspond to the related conditioning. A more quantitative approach would be to employ the Fréchet Inception Distance (FID) (↓) [156] which effectively captures how different the features extracted are between the ground truth dataset and that predicted by the model - effectively capturing the distance between two distributions of embeddings. Originally for image GANs, it has been applied to 3D grasp evaluation by embedding point clouds or rendered images. Given a set of real samples $\{r_i\}$ and generated samples $\{g_j\}$, extract feature vectors f_r and f_g using a pretrained network (e.g. Point-E or Inception). Let μ_r, Σ_r be the empirical mean and covariance of $\{f_r\}$, and similarly μ_g, Σ_g for $\{f_g\}$. Then the FID is

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 \; + \; \mathrm{Tr}\big(\Sigma_r + \Sigma_g - 2(\Sigma_r\,\Sigma_g)^{1/2}\big).$$

In grasping, one computes P-FID by embedding full hand–object point clouds and I-FID by embedding rendered grasp images. Lower FID means the generated grasp distribution is closer to the reference distribution. FID thus captures both accuracy and diversity of grasp shapes as a whole. One can apply this directly between the predicted and ground truth point clouds using features extracted from a model like Point-e which generates 3D point clouds from text prompts [155]. This axis of comparison can be yet another way to leverage the common-sense reasoning of LLMs and VLMS. In SemGrasp [91], for example, the rendered image and the text-prompt conditioning fed as input are given to GPT-4 and asked to score the semantic consistency between 0 and 100.

2.3 Diversity

For the generative models which are needed for grasp synthesis, it is important to measure the diversity of the grasps produced. For works that focus on the physical stability of the grasps, it's often noticed that optimizing for less inter-penetration between hand and object reduces the diversity of the grasps produced, so it's often an important issue to address [90], [84], [75]. Generation diversity is also crucial when deploying these grasps in the real world as oftentimes multiple options need to be tried in order to successfully grasp an object. For works that condition the grasps on a prompt, if the prompt is ambiguous, diversity is important to measure to sanity check that the model doesn't degenerate to an example from the dataset. A straightforward way is to measure the standard deviation (↑) of the joint angles and global translation and rotation parameters. Other works first cluster the grasps produced into clusters and then report the average size of the entropy of the clusters (↑). For example, given n grasps for the same object, one may compute

$$\sigma_{\text{trans}} = \frac{1}{n}\sum_{i=1}^{n}(t_i - \bar{t})^2, \quad \sigma_{\text{rot}} = \frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2, \quad \sigma_{\text{joint}} = \frac{1}{n}\sum_{i=1}^{n}(q_i - \bar{q})^2,$$

where t_i, r_i, q_i are respectively the translation magnitude, rotation angle (or each Euler component), and a representative joint-angle of grasp i. Higher standard deviation means grasps vary more. To capture multi-modality, cluster the generated grasps (e.g. by K-means on their keypoints) and compute the entropy of the cluster assignment distribution. If there are K clusters with counts n_k, and $N = \sum_{k=1}^{K} n_k$, the cluster entropy is

$$H = -\sum_{k=1}^{K} \frac{n_k}{N} \log\!\left(\tfrac{n_k}{N}\right).$$

A higher H (up to $\log K$) indicates more uniform usage of modes, hence greater diversity.

2.4 Functionalities

Other than the three main axes of comparison listed above, there are properties that can be very valuable when it comes to dexterous robot grasp synthesis. Especially when looking to deploy in the real world, models that can predict grasps for multiple robot embodiments given information or a point cloud of the robot hand can allow the model to be deployed regardless of embodiment available. Some works like D(R,O) [24] can predict grasps even when the point-cloud fed of the object or hand is partial, which is useful as it's rare to have accurate and complete point clouds with modern depth cameras. Another real-life metric that's important is inference time, which is useful for trying different grasps in a reasonable amount of time or predicting stable grasps given real-time disturbances. For conditioned grasp models, a human interface metric to keep in mind is the ease of providing the conditioning prompt (is it easy to generate or does it require extra steps or rigorous templating?).

3 Grasp Datasets

Dexterous robotics hands are fundamentally different from that of parallel-jaw grippers: each grasp must reason over dozens of joint angles, rolling contact, and task semantics. As a result, benchmark corpora must be not only *large* but also *rich* → capturing multi-finger contact geometry, diverse object categories, and, increasingly, language or affordance labels. In the remainder of this section we catalogue the principal public datasets, describe how they are synthesized or captured, and comment on the modeling biases they introduce.

Many modern approaches to dexterous grasping rely on deep learning, which demands large-scale, diverse datasets to achieve robust and generalizable performance. To meet this need, data is typically acquired

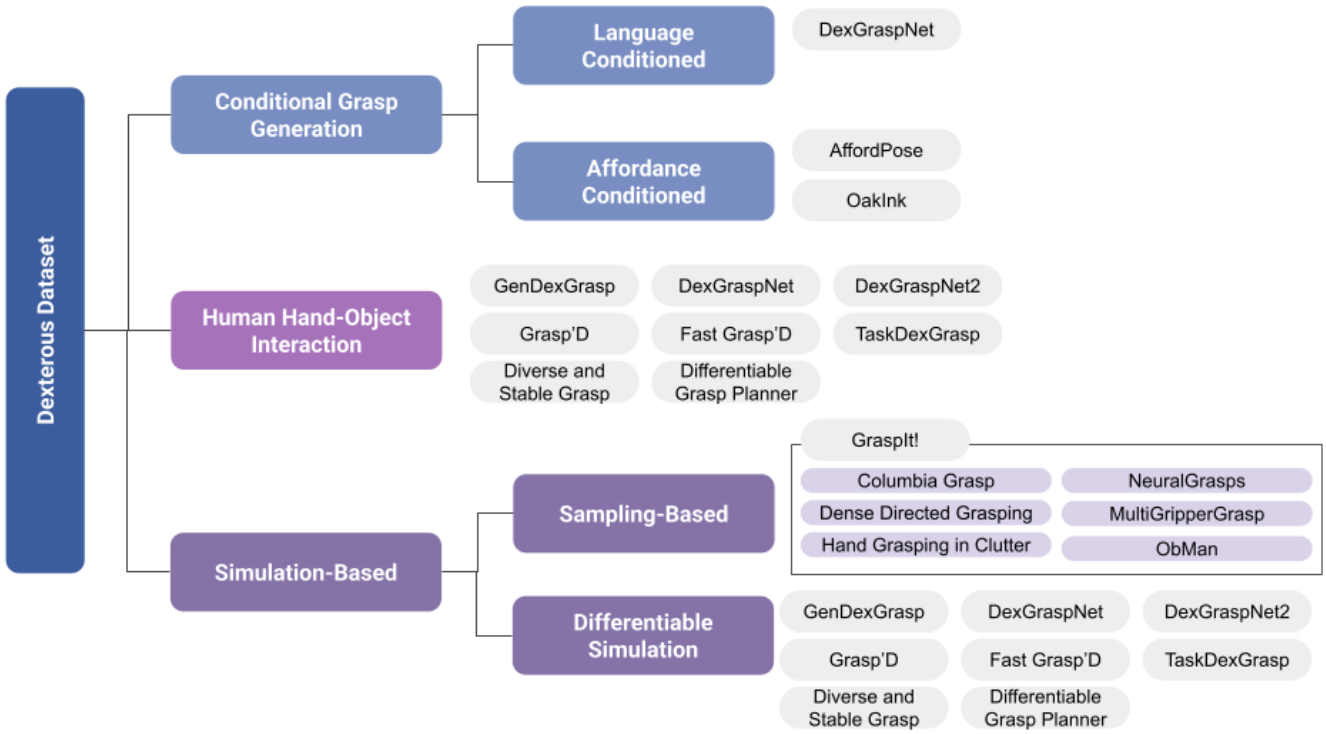


Figure 2: Dataset Hierarchy Figure

in one of two ways: (i) by optimizing grasps in simulation or (ii) by observing real-world hand–object interaction.

3.1 Simulation-Based Datasets

Simulation corpora fall into two families. *Sampling planners* rely on stochastic search and analytic metrics (e.g. force closure), whereas *differentiable simulators* embed the metric in an automatic-differentiation loop that scales to millions of grasps per hour.

3.1.1 Sampling Based Simulation Methods

GraspIt! [52] is a versatile simulator designed for generating and evaluating grasps given arbitrary hand and object models. It has been used to produce multiple datasets, including the Dense Directed Grasping dataset [55], the Hand Grasping in Clutter dataset [58], the NeuralGrasps dataset [112], the MultiGripperGrasp dataset [57], the Columbia Grasp Dataset [54], and the ObMan dataset [56], for training and benchmarking grasp synthesis models. In order to generate data, users can leverage one of GraspIt!'s built-in grasp planners to sample diverse grasp poses around the object. However, the majority of these samples do not produce quality grasps, so the candidate grasps can be filtered based on GraspIt!'s analytic metrics (e.g. force closure, epsilon quality) or custom heuristics.

3.1.2 Differentiable Simulation Methods

While GraspIt! and other sampling-based techniques have proved to be useful tools, they are inefficient for dexterous hands due to the high dimensionality of the search space. Although dimensionality reduction techniques like eigengrasps [53] have been used to make the problem more tractable, this simplification restricts the diversity and expressivity of generated grasps. Due to these limitations, there has been a shift towards differentiable grasping in recent years. These approaches (Differentiable Grasp Planner, Diverse

and Stable Grasp, Fast Grasp'D, Grasp'D, DexGraspNet, DexGraspNet2, GenDexGrasp, TaskDexGrasp) [47, 48, 42, 41, 44, 45, 43, 46] use differentiable simulation to define grasp metrics that enable gradient-based optimization of grasp poses. More information at 4.3.

3.1.3 Design Trade-Offs

Despite wide adoption, random search in a greater than 20-DoF space is sample-inefficient; quality filters discard 90–99 % of proposals, and mesh inter-penetration remains common. Differentiable corpora alleviate these issues, producing higher-stability, contact-realistic grasps an order of magnitude faster. However, they inherit simulator bias—most notably from Isaac Gym's contact and friction models—and currently cover fewer exotic object categories than their sampling counterparts. Future datasets will likely hybridism both paradigms: using differentiable seeds to guide targeted sampling in under-represented shape families, while injecting real-world system identification to curb simulation bias.

3.2 Human Hand-Object Interaction Datasets

Outside of simulation, many works [60, 65, 66, 73, 63, 69, 71] have focused on collecting real-world hand-object interaction data, including bimanual data [72, 70, 62]. Although some of these datasets were originally intended for other tasks such as hand-pose estimation or hand-object generation, they can be repurposed for dexterous grasping. While this approach might lack the scale of simulation methods, it offers human-like grasp demonstrations, functional grasping behaviors, and the ability to capture both static grasp poses and pre- or post-grasp motion.

3.2.1 Marker–Based Motion Capture Corpora

Early work relies on optical MoCap to obtain millimeter accurate meshes of both the hand and the manipulated object. *GRAB* records whole body interactions of ten subjects grasping 51 everyday objects, yielding 1.3 M high resolution meshes with per vertex contact labels cite.

3.2.2 Egocentric RGBD Video Corpora

To expose algorithms to real–world clutter and self–occlusion, several corpora adopt a first–person camera. *HOI4D* contains 2.4 M RGBD frames across 4 000 sequences, capturing 800 object instances manipulated. *H2O* supplements egocentric colour–depth footage with two hand 3D pose and per frame interaction labels, enabling research on bimanual manipulation. The very recent *HOGraspNet* pushes scale further by covering the full 33 grip taxonomy across 1.5 M annotated frames from 99 participants. Although annotation noise increases compared with MoCap sets, egocentric corpora better reflect daily manipulation scenes.

3.3 Datasets for Conditional Grasp Generation

There have been recent efforts [74, 89] focused on conditional grasp synthesis in order to generate context-aware grasps, with conditioning information including affordances and language. To support this, several datasets have been developed or extended to include contextual information alongside grasp annotations. Affordance datasets, such as AffordPose [64] and OakInk [61], are typically real-world hand-object datasets where subjects are either instructed to grasp an object with a specific intent or later asked to label the resulting affordance. Similarly, for language-conditioned grasping, existing datasets like DexGraspNet[44] have been augmented with natural language prompts using multimodal large language models (MLLMs) such as GPT-4o [115] to integrate semantic context directly into grasp synthesis.

Dataset (Year)	Type	Hand	#Objects	#Grasps/Data	Modalities	Eval. Annotations	Unique Features
DexGraspNet (2022)	Sim	ShadowHand	5,355	1.32M grasps	None (poses only)	Force-closure (optimized)	Very large simulated dataset; validated via an Isaac Gym checker
DexGrasp Anything (2024)	Hybrid (Real+Sim)	ShadowHand (+ human)	15,698	3.40M grasps	Meshes, 6-DOF poses	Force closure, penetration	Largest-to-date dexterous dataset; curated and physics-expanded
RealDex (2024)	Real	ShadowHand	52	59K grasps	RGB-D (multi-view)	Human annotation (quality)	Teleoperated dexterity; multi-view capture; human-like patterns
MultiDex (2023)	Sim	ShadowHand	58	16K grasps	None	Optimized grasps	Small-scale high-quality set for dexterous pre-training
UniDexGrasp (2023)	Sim	ShadowHand	5,519	1.12M grasps	None	Optimized grasps	Million-scale dataset; 200+ grasps per object
GRAB (2020)	Real	Human (MANO retarget)	51	1.64M grasps	RGB-D (multi-view)	Human grasp labels	Real human grasps; learn hand–object interactions
HO3D (2021)	Real	Human (MANO)	10	77K frames	RGB-D (hand+object)	6D hand+object poses	Egocentric videos; synchronized 6D pose annotations
DexYCB (2021)	Real	Human (MANO)	20	582K frames	RGB-D	6D hand+object poses	Multi-camera capture; dense annotations

Table 2: Summary of key dexterous grasping datasets, listing publication year, data source (simulation, real, or hybrid), hand model, object and grasp counts, data modalities, evaluation annotations, and distinguishing features (e.g., large-scale simulation, multi-view capture, human demonstrations). These corpora enable both scalable training in simulation and transfer to real-world dexterous manipulation.

4 Learning to Predict Grasps

Generating stable, dexterous robot grasps is challenging due to the high-dimensional action space, variety of objects, and the multi-modal nature of grasp distributions. Recent methods tackle this by leveraging large-scale grasp datasets, inspired by data-driven successes in natural language processing and computer vision.

4.1 Grasp Representation

The grasp representation used to inform the model during training is critical to the efficiency and accuracy of generative grasp models. An effective grasp representation should capture grasp diversity, allow precise finger placement, and reduce ambiguity to support the optimization of stable, constraint-satisfying robot joint angles. Representations play a central role not only in defining the model's output space but also in shaping its input conditioning. While the following sections primarily describe representations used to parameterize the output grasp, many of these can also be adapted as conditioning inputs — providing rich, structured guidance that informs and constrains the grasp synthesis process.

Contact Maps: Contact map methods explicitly indicate potential contact regions on an object using heat maps. These maps are typically represented either as a probability distribution along the surface of an object or as an estimate of penetration depth between the hand and the object. However, they do not fully resolve the detailed configuration required for a grasp, leaving unanswered questions about the exact placement of each finger and the corresponding direction of contact. These ambiguities include not only

where contact should occur but also which part of the hand should make contact and how that contact should be oriented. ContactGen [80] tackles these questions by generating three sequential contact maps to encode additional information. The first is a heat map predicting likely contact regions on the object; the second assigns each point to a corresponding hand part; and the third specifies the direction of contact, representing the normal vector at the contact point aligned with the contacting hand surface. Other works [29, 77, 81] use the contact map as a guide for a later optimization process to refine the initial grasp, gradually resolving finger assignments and contact orientations to produce plausible, stable grasps.

Hand Parameter Prediction: Parameter prediction approaches directly predict the position, orientation, and joint angles of the hand. Although learning is simplified and training is sped up due to the reduced output space, the lack of geometric information involved requires these methods to optimize or filter the grasps further. After prediction, forward kinematics is typically used to generate the full hand mesh from these parameters, whose form may vary depending on the hand model. This approach significantly reduces computational complexity compared to methods that predict dense contact representations. Parameters can be tokenized and auto-regressively predicted leveraging large language models, which can easily be conditioned on extra information like point clouds and text annotations [91, 89, 84, 88]. For example, SemGrasp [91] decomposes the hand configuration into interpretable tokens labeled orientation, manner, and refinement—capturing progressively detailed semantic priors about the grasp. Multi-GraspLLM [89] uses a binning scheme over hand parameters, alongside a dedicated "hand token" that allows for generalization across hand models. Tokenization enables multimodal conditioning via a shared embedding space, where text, images, and point clouds are embedded alongside grasp tokens. Other approaches generate parameter vectors directly using diffusion-based or simulation-informed models, which are later refined for better stability and alignment.

Geometrically Aware Representations: Geometrically aware methods introduce inductive biases that enable models to directly incorporate intricate object geometries during grasp generation and optimization. For instance, G-HOP [92] defines Hand-Object Interaction (HOI) using an "interaction grid," formed by concatenating the object's signed distance field (SDF) and the hand's skeletal distance field—a 15-channel volumetric grid where each channel corresponds to distances from a hand joint. This skeleton-based encoding allows diffusion models to better reason about spatial hand structure during denoising, especially when conditioning on heterogeneous inputs like text. After denoising the grid, fast differentiable optimization can be used to extract hand parameters and object point clouds. Similarly, $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp [24] represents HOI as a matrix where each element denotes the distance between a point on the object and a point on the hand point cloud. This matrix serves as a mid-level representation that bridges object-centric and hand-centric encodings, enabling faster inference without requiring an additional optimization step. By leveraging multilateration to compute the hand point cloud from object points, this method achieves embodiment invariance and robustness to partial point cloud observations, which is valuable for real-world deployment. In addition, GraspingField [93] offers an implicit representation of contact via a deep network that maps every 3D point to signed distances with respect to the object and hand, along with hand part labels. The contact manifold is found by identifying points where the field output is zero, and interpenetration can be measured by summing the field's negative values—providing a principled and geometry-aware way to score and optimize grasps.

Extra Conditioning Representations: Grasp synthesis can be guided by conditioning on additional inputs that act as inductive biases, enabling models to generate grasps that are more accurate—e.g., with reduced hand-object interpenetration—and more aligned with specific tasks or contexts. One approach involves encoding hand-object interaction explicitly through "touch codes," which assign bit-level labels to object point cloud regions based on which segmented parts of the hand should contact them. In works such as Toward Human-Like Grasp [74, 94], this representation is extended with a 4-bit "intention" code to capture the nature of the contact—such as pressing, sliding, or clicking—allowing the grasp to reflect human-like functional priors. Other methods like RegionGrasp [75] directly condition the model on specific regions of the object point cloud, using these regions as a spatial mask to bias the model toward contacting selected areas.

To enable task-oriented or user-guided grasping, natural language is often used as a conditioning modality. However, language is inherently ambiguous—multiple grasps can satisfy the same verbal instruction—making it challenging for models to learn robust mappings from free-form descriptions. To mitigate this, some approaches constrain the language space using structured prompts and train on curated datasets with consistent annotation formats [88, 90, 87, 95]. Others propose hierarchical annotation schemes, separating high-level semantic intent from low-level contact details [89, 91]. In all cases, large language models (LLMs) and vision-language models (VLMs) are leveraged to synthesize additional training annotations, enabling broader generalization and more nuanced grasp behaviors through rich, multimodal supervision.

4.2 Backbone Methodologies

Beyond representation, methods vary in the deep learning architectures and learning policies chosen for grasp synthesis. This choice depends on the conditioning inputs, desired outputs, and task requirements. While many focus on a single unified architecture, some combine paradigms to leverage state-of-the-art models and handle diverse data.

Transformers: Transformers are a convenient and performant choice for methods with heterogeneous inputs, such as when the grasp is to be conditioned on additional information that can be tokenized. Some works train self- and cross-attention layers from scratch to process input embeddings, while others use pretrained LLM or VLM backbones and fine-tune them using parameter-efficient techniques such as LoRA [25], which enables leveraging pretrained knowledge without retraining full models. These models are especially suited for multimodal inputs—such as text, point clouds, or images—where inputs can be embedded and fed into attention modules. In Multi-GraspLLM, for example, point cloud features are adapted into the same token space as text, and hand type tokens are appended to the sequence before being decoded by the transformer to generate hand parameters [89]. However, decoding continuous outputs like hand parameters from discrete tokens often leads to low physical accuracy. To mitigate this, many approaches introduce a downstream refinement step—such as optimization, filtering, or additional diffusion layers—to improve realism and contact fidelity [88, 87]. Transformers thus serve as both flexible input encoders and output decoders, especially when paired with rich, tokenizable representations.

Diffusion/Flow Matching: Generative methods like diffusion [120, 98] and flow matching [149, 148] are widely used for grasp synthesis due to their ability to model complex, multi-modal distributions—enabling the generation of diverse, yet feasible grasps. The stochasticity introduced in diffusion's denoising steps improves diversity while maintaining quality, and has shown advantages over earlier VAE-based approaches in stability and physical plausibility [84]. A common choice is to perform denoising directly in the hand parameter space, but this lacks geometric awareness of the hand-object interaction. To address this, many works condition the generation process on representations like contact maps or affordance cues. For instance, ClickDiff [99], NL2Contact [76], and AffordDexGrasp [100] use contact-based conditioning to guide the diffusion toward feasible grasps. Some works diffuse in latent spaces (e.g., UGG's shared object-hand embedding), requiring auxiliary steps such as physics-based filters or optimization to enforce realism [98]. Classifier-free guidance is often used to enable flexible conditioning, with CLIP and point encoders processed via cross-attention layers to bridge modalities before feeding them to the base architecture—typically an MLP, U-Net, or Transformer.

Autoencoders: Autoencoders are widely used in grasp synthesis to compress high-dimensional data and generate grasps in a structured latent space. Variational Autoencoders (VAEs) are particularly popular for generating grasps from compressed embeddings of hand-object geometry [78]. These models often integrate auxiliary losses—such as penalties for interpenetration or instability—to improve the quality of decoded grasps. UGG [98] combines embeddings of the object, hand, and contact anchors to guide diffusion through an autoencoded latent space. RegionGrasp [75] uses autoencoders to fuse region-specific conditioning with hand pose embeddings. Hierarchical and discrete variants—like VQ-VAEs and DQ-VAEs—further help capture the categorical structure of grasps. For example, GrainGrasp [79] predicts a contact map from a CVAE and optimizes hand pose to match it, while SemGrasp [91] learns orientation, manner, and

refinement tokens via hierarchical VQ-VAEs. ContactGen [80] similarly decomposes contact into three maps (contact location, part, direction), using separate CVAEs for each. These structured approaches not only simplify learning but also enable modular grasp editing and better generalization across tasks.

Other Architectures: Though less common than diffusion or autoencoder-based methods, generative adversarial networks (GANs) have been used in grasp synthesis to enforce realism and physical plausibility through adversarial training. In DexGANGrasp [175], a discriminator evaluates sampled grasps and helps the generator focus on those most likely to succeed in real-world execution. Similarly, GanHand [174] uses a discriminator to enforce anthropomorphic realism for human hand grasps. In cases where the grasp representation is especially informative—such as Touch Codes—simpler architectures like CNNs or MLPs can suffice. For instance, in Toward Human-Like Grasp [74] and FunctionalGrasp [94], hand-object segment alignments are encoded and directly regressed via cascaded MLPs, followed by optional refinement steps. These approaches highlight that, with strong representations, even non-generative models can produce high-quality, context-aware grasps suitable for downstream deployment.

4.3 Optimization and Refinement

Additional loss terms or filtering steps are often employed to ensure grasps are physically plausible and stable. Most approaches first generate a coarse grasp and then refine it for stability. For methods that incorporate external conditioning such as text prompts, contact maps, or other task-specific cues [88, 90, 76], the refinement stage is often decoupled from the initial synthesis process. This separation allows the system to strike a trade-off between adherence to conditioning inputs with the optimization of physical stability.

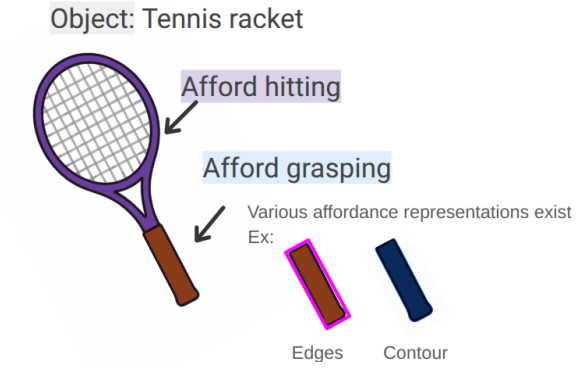
Metrics as Losses: Metrics like Chamfer Distance, hand-object interpenetration volume, self-penetration, and the L2 norm of hand parameters are commonly used as auxiliary losses during grasp optimization or generation due to their differentiability and computational efficiency (see section 2 for detailed definitions). These metrics are especially suited for supervision when the model predicts hand point clouds or joint parameters, as they enable stable gradients and interpretable improvements in grasp quality. Beyond general-purpose losses, several works incorporate task- or structure-specific terms. Losses are then designed to attract the corresponding hand parts to the appropriate object regions while repelling others — thus encouraging both functional contact and reducing self-penetration through the same repulsion mechanism. Works that use contact map conditioning often apply additional losses to align predicted contacts with those expected from the conditioning signal. Because these loss terms interact in non-trivial ways, their relative weighting requires careful tuning and often depends on the initialization quality or grasp representation. Empirical studies have shown that the effectiveness of specific losses is method-dependent, and optimization performance can degrade if the coefficients are poorly balanced.

Filters/Discriminators: Many generative grasping methods produce multiple candidate grasps for a given input, which are then filtered to select the most stable and task-appropriate option. A common approach is to compute physical or geometric quality metrics—such as interpenetration, force closure, or contact area—for each candidate and rank them accordingly. In simulation-based pipelines, these grasps may be evaluated in a physics engine to test their robustness under external perturbations. For instance, DexTOG [87] simulates each grasp and further refines its evaluation using a reinforcement learning policy conditioned on natural language to determine whether the grasp is functional. However, physics-based filtering is computationally expensive, so most methods either limit the number of candidates (typically 5–10) or use truncated simulations. For language-conditioned grasping, filtering may also involve vision-language models (VLMs). A rendered image or point cloud of the grasp is passed to a VLM along with the text prompt to score how well the generated grasp matches the intent. In DexGraspDiffusion [84], a functional discriminator is trained by encoding the grasp's point cloud and comparing it with a CLIP-based embedding of the instruction. This similarity score guides the selection of the best-aligned grasp among the candidates. These filtering mechanisms, though auxiliary, often determine whether the generated grasps are suitable for real-world deployment, especially under task-specific or semantic constraints.

Energy-Based Models: Some works synthesize stable grasps directly through optimization over an energy landscape, rather than training predictive models. In Synthesizing Dexterous Grasps via Differentiable Force Closure Estimator [48], the traditional force closure metric is reformulated into a differentiable function that estimates how robust a contact configuration is to external wrench forces. This energy is combined with terms for self-penetration and interpenetration to define a full energy landscape over hand parameters. The optimization begins from a random hand pose and uses MALA (Metropolis-adjusted Langevin algorithm), a sampling-based method well-suited for navigating complex landscapes, to find stable configurations. Notably, this approach is morphology-agnostic and works across various robot hands. DexGraspNet [44] builds on this method with an improved optimization process for scalable grasp generation. By initializing the hand in an open state and replacing MALA with simple gradient descent (made feasible by a smoother energy landscape), it enables faster synthesis without sacrificing grasp quality. Penetration losses are computed using signed distances between the hand and object point clouds, ensuring smoother gradients even with non-thick object meshes. GrainGrasp [79] further extends this by conditioning the energy landscape on a predicted contact map, guiding the optimization not just toward stability, but also toward semantic alignment with task or intent-level grasp constraints.

5 Grasp Execution

Dexterous grasp execution represents the final and most critical phase in robotic grasping pipelines. Unlike simple two-finger grippers used for pick-and-place, multi-fingered anthropomorphic hands can exploit rich object affordances (e.g., handles, edges, contours) to achieve stable, functional grasps in unstructured settings. As tasks have shifted from rigid pick and place toward in-hand reorientation, tool use, and fine manipulation [32], traditional model-based controllers have struggled to handle complex contact dynamics and sensor uncertainty [1]. Grasp execution is not merely a pose to reach but a trajectory to realize, shaped by continuous control updates and responsive behaviors. The growing need for generalization and robustness in unstructured settings has driven research toward learning-based methods that can fuse multimodal inputs (vision, tactile, proprioception) and accommodate partial observability.



This example illustrates object affordances using a tennis racket. The racket head affords hitting, while the handle affords grasping. Different representations such as edges or contours are used to encode these affordances, enabling semantic and functional understanding during grasp planning.

Figure 3: Affordance representations on a tennis racket illustrating grasping and action-oriented regions.

5.1 General Problem Formulation

The fundamental challenge of dexterous grasp execution lies in reliably transitioning a hand–object system from an unconstrained initial state to a stable interaction state under uncertainty. In dexterous grasp execution, our goal is to take a multi-fingered hand from an initial, unconstrained state and drive it into a stable, functionally adequate grasp of the object. Formally, let $\mathcal{X}$ be the continuous state space of the combined hand–object system (e.g., joint angles, object pose, tactile readings) and let $X_G \subset \mathcal{X}$ be the subset of successful grasp states where the object is held securely and ready for downstream manipulation.

Figure 4: Representative examples of robotic and human hands used in dexterous manipulation research. The figure includes both anthropomorphic designs (e.g., Allegro, Shadow, LEAP, MANO) and task-specific grippers (e.g., Barrett, Robotiq, D'Claw).

A closed-loop control policy $\pi : \mathcal{X} \to \mathcal{U}$ maps each observed state $x_t \in \mathcal{X}$ to an action $u_t \in \mathcal{U}$ (e.g., joint-space torques or Cartesian fingertip velocities), such that over a horizon T the system trajectory $x_0,\ x_1,\ \ldots,\ x_T$ satisfies $x_{t+1} = f(x_t,\ u_t),\ x_0$ given, $x_T \in X_G$.

5.2 Control Paradigms

In addition to learning-based approaches, there exist many classical grasp execution methods that rely on hand-designed control logic and analytical models to plan and adjust grasp trajectories using explicitly programmed rules and feedback. Broadly, grasp execution methods can be divided into two families based on how they generate and use feedback during execution:

1. **Classical methods** - Classical grasp execution methods rely on hand-designed control logic and analytical models to plan and adjust grasp trajectories using explicitly programmed rules and feedback. Each of the methods vary in how and when sensor feedback is incorporated (See Section 5.3)
2. **Learning-Based Execution Policies** - In contrast, Learning-Based Execution Policies use data-driven techniques (e.g., reinforcement or imitation learning) to acquire a closed-loop grasping policy from demonstrations or trial-and-error (See Section 5.4)

Each of these subcategories can be viewed as a specialization of the general control model, differing primarily in how $\pi(x)$ is specified and how feedback is incorporated during execution.

Within learning-based approaches, we can categorize them into three main paradigms:

Model-Free Reinforcement Learning (Sections 5.4.3) learns policies (e.g., PPO, SAC, TD3) by maximizing task-specific rewards through repeated interaction.

Curriculum & Residual Learning (Sections 5.4.5) structures training over time by progressively increasing task complexity (curriculum) and refining base policies through incremental adjustments (residual learning).

Imitation Learning & Distillation (Sections 5.5) directly imitate expert trajectories (behavior cloning, DAgger) or distill privileged "teacher" policies into vision-based "students."

Each paradigm implements $\pi(x)$ differently, either by optimizing a reward (RL), mimicking demonstrations (IL), blending both (hybrid), or leveraging structured strategies such as curriculum learning and residual learning, all produce closed-loop controllers that react online to sensor feedback. The details of these methods appear in Section 5.4.

5.3 Other Execution Approaches

In dexterous grasping, the classical execution strategies can be grouped into three interconnected paradigms:

5.3.1 Open-Loop Execution

Executes a fully precomputed trajectory without runtime corrections. All decisions—approach path, finger timing, final joint configuration—are determined offline using the best available object model. During execution, low-level joint controllers simply track the planned trajectory, but no higher-level adjustments occur. Examples include computing a single grasp pose from vision [158, 159, 160] and then executing it blindly. Formally, if

$$x_{t+1} = f(x_t, u_t), \quad x_T \in X_G, \quad u_t = \mu_t \ \ (t = 0, \ldots, T),$$

with $\{\mu_t\}$ obtained offline to minimize $J(x_T)$, then no feedback is used during execution. Open-loop methods assume a static, well-modeled environment and are computationally efficient [170, 162], but they are brittle: unexpected contacts or object perturbations cannot be corrected, limiting robustness under uncertainty [161, 163, 168, 164, 169].

5.3.2 Reactive Execution (Event-Driven Feedback)

Augments a nominal trajectory with simple, sensor-driven adjustments. Instead of executing blindly, the controller transitions between discrete phases (e.g., "approach," "adjust," "close") based on events like fingertip contact. Concretely,

$$u_t = \pi_{s_t}(x_t), \quad s_{t+1} = F(s_t, x_t),$$

where s_t denotes the current phase and F triggers transitions when sensors (e.g., tactile or torque thresholds) detect contact. Early work [168] used fingertip sensors to correct unexpected collisions or adjust grip force; more recent methods integrate high-speed force/proximity sensing [166, 167] or CNN-based slip prediction [165] to trigger corrective regrasp maneuvers. Reactive strategies improve robustness over open-loop at relatively low complexity but rely on hand-tuned heuristics and can fail under unanticipated or compound disturbances.

5.3.3 Model-Based Feedback Control

Treats grasp execution as a continuous control problem, actively regulating motion and contact forces using sensor feedback and a mathematical hand–object model. Typical controllers include:

Hybrid Position/Force Control: Separates unconstrained (position) and constrained (force) directions via a selection matrix.

Impedance Control: Implements a virtual spring–damper behavior, blending position tracking with compliant force regulation.

Model Predictive Control (MPC): Solves a short-horizon optimization at each timestep to minimize tracking errors while respecting contact constraints.

Learning-Augmented Models: Incorporates learned components (e.g., neural network estimators) into a model-based loop to handle unmodeled dynamic.

Model-based methods can guarantee stability and handle continuous uncertainties (e.g., slight pose errors) more gracefully than discrete reactive rules, but they require accurate dynamics models, precise sensors, and substantial computation. Tuning these controllers for complex, high-DOF hands remains challenging.

Despite their individual merits, open-loop, reactive, and model-based controllers each exhibit trade-offs that become increasingly problematic as dexterous grasping tasks grow in complexity or are deployed

in unstructured environments. Open-loop executions are fast but brittle; reactive methods recover from only limited disturbances; and model-based controllers require precise models and a heavy computation. Learning-based execution policies, in contrast, aim to acquire closed-loop control strategies directly from data, through reinforcement learning or imitation learning, thereby capturing nuances of contact dynamics and environmental variability without requiring explicit modeling of every physical parameter.

5.4 Execution Methods

Learning-based methods have rapidly emerged as a powerful alternative for dexterous grasping and manipulation. Rather than depending on hand-designed controllers or carefully tuned models, these methods "learn from experience"—either through trial-and-error (Reinforcement Learning) or by mimicking expert demonstrations (Imitation Learning). By directly leveraging data, learning-based approaches can uncover complex strategies for finger coordination, contact forces, and adaptation to sensor uncertainty.

5.4.1 Defining the Grasping Task and Intention

Defining the grasping task and intention involves identifying the core objectives and mapping them onto a sequence of actions that ensure reliable and task-specific interactions. At its foundation, the grasping task typically encompasses three primary objectives:

1. **Pick:** Approaching and securing the object by selecting appropriate contact points.
2. **Hold:** Maintaining a stable grasp that prevents slippage, which is critical for carrying or manipulating the object.
3. **Manipulate:** Executing task-specific motions—such as reorienting, assembling, or transferring the object—by adapting the grasp throughout the manipulation process.

A key consideration in this definition is the balance between stability and task-specific demands. For instance, a robust grasp that excels in holding heavy objects might not be suitable for delicate manipulation tasks where compliance and fine control are essential. Thus, the grasp strategy must account for both mechanical stability (ensuring force closure and resistance to perturbations) and the specific requirements of the intended manipulation.

5.4.2 Observation and Action Representations

An effective grasp execution tackles not only the choice of control paradigm but also on how the robot perceives the world and issues commands. In learning-based methods, both observations and actions must be represented in a way that captures the rich, contact-rich nature of dexterous manipulation. Here we briefly review the most common observations modalities and action spaces within modern exectuion policies.

Observation Modalities Policies $\pi(x)$ require sensory inputs that contain the state of both hand and object. There are two main modalities dominate current research: Vision, Tactile & Proprioception.

Vision: RGB-D & Point Clouds Vision is the primary modality for most robotic grasping systems. RGB-D cameras provide both color and depth, allowing methods to infer 3D geometry of objects. Many learning-based grasp policies use depth data to construct point clouds of the scene, which serve as rich inputs for neural networks [159, 197]. By operating on 3D point sets, the policy can directly perceive object shapes and positions in space. The advantage of depth-based vision is evident in closed-loop systems where one

Type	Representation	Description	Examples & Use Cases
Observation	Vision (RGB-D)	Provides global context, object shape, and pose from camera data.	Initial grasp planning, object detection, avoiding obstacles.
	Proprioception	Internal measurements of the robot's state	Essential for closed-loop control; informs the policy of hand's configuration (e.g. joint angles/velocities).
	Tactile	Contact-based feedback on finger-mounted sensors	Detecting slip, confirming contact, fine-grained manipulation.
Action	Joint-Space	Command joint positions, velocities, or torques for each DoF.	Offers maximum flexibility; used for complex in-hand reorientation.
	Cartesian-Space	Command the 6D pose (position + orientation) of end-effector/fingertips.	Intuitive and lower-dimensional; good for reaching and pre-grasp alignment.
	Latent-Space	Command a low-dimensional vector that decodes into a motion primitive.	Improves sample efficiency via encoded constraints; used in hierarchical policies.

Table 3: Overview of the primary observation (input) and action (output) representations used in learning-based dexterous grasp execution. Observation modalities (vision, proprioception, tactile) supply rich state information to the policy, while action spaces (joint-space, Cartesian-space, latent-space) define how commands are issued; example use cases illustrate typical applications of each representation.

can demonstrate real-time reactive grasping by generating grasps from depth images at each timestep [163]. Likewise, large-scale deep RL policies such as QT-Opt learned to grasp using only monocular images by scaling up real-world data collection

Tactile & Proprioception While vision provides global context, tactile sensing offers local, high-resolution feedback once contact is made. Tactile sensors (e.g. GelSight [191], DIGIT [152]) mounted on gripper fingers can detect contact location, pressure, and incipient slip, enabling policies to adjust grasps in real time [192]. Integrating touch has been shown to significantly improve grasp reliability [193, 194]. Their deep multimodal model predicts grasp outcomes and chooses adjustments, achieving higher success with gentler forces than vision alone. The addition of tactile feedback allows the robot to feel object slip or instability and react accordingly [195], which purely visual policies cannot easily do once the object is occluded by the gripper. In parallel, proprioceptive sensing is a vital modality included in almost all control policies. Proprioception informs the policy of the arm's current configuration relative to the target. Many frameworks [171, 186, 187] also append the end-effector pose or a goal pose to the observation, which can guide the policy toward a desired grasp configuration.

In summary, modern grasping systems tend to be multi-modal: vision provides external object data, proprioception supplies the robot's state, and tactile sensors close the loop with direct contact feedback. Beyond these primary modalities, systems may also incorporate in-hand force–torque sensors, magnetic sensors, or "self-sensing" objects equipped with onboard IMUs or force–torque measurements. This rich observation space helps policies handle occlusions and uncertainties during grasp execution.

Action Spaces The choice of action representation significantly impacts the sample efficiency and performance of learned grasping policies. We categorize action spaces into two fundamental types: Joint-Space Commands and Cartesian & Latent Commands.

Joint-Space Commands One straightforward way to represent actions is in terms of the joint configuration of the robot [189, 190].In this approach, policies output desired joint positions $\mathbf{q}_t \in \mathbb{R}^n$, velocities $\dot{\mathbf{q}}_t$, or torques $\boldsymbol{\tau}_t$ for each degree of freedom. While this offers maximal control flexibility - enabling any kinematically feasible pose - the high dimensionality (e.g., 24D for Shadow Hand) creates exploration challenges [32]. Early model-free RL work like [197] demonstrated joint torque control could achieve dynamic

dexterity, but required careful reward shaping and extensive simulation training. The dimensionality issue compounds for bimanual systems or mobile manipulators, where action spaces may exceed 30 DoF [21].

Cartesian & Latent Commands Modern systems therefore typically avoid pure joint-space learning except for specialized tasks like in-hand reorientation [193], instead using eitherjoint-space controller with higher level Cartesian, latent inputs [83, 21, 20, 140] or hybird representations that mix joint and task-space control [125, 199, 197]. Rather than operating directly in joint space, policies may output either: (1) end-effector/fingertip poses in task space ($SE(3)$ coordinates) decoded through learned inverse kinematics [83, 21], or (2) compressed latent vectors that parameterize coordinated motion primitives [114, 20]. This hybrid paradigm offers three key advantages: first, the reduced dimensionality (typically 3-10D for latent spaces vs. 20+DoF for anthropomorphic hands) greatly improves sample efficiency [197]; second, Cartesian targets maintain physical interpretability for reward shaping and safety verification; third, latent spaces can encode implicit complicated contact constraints. However, these benefits come with added complexity in training dynamics: the policy must learn to generate latent codes that decode to kinematically feasible motions, often requiring auxiliary losses on reachability or collision avoidance [125]. Recent systems like [198] further bridge this gap through hierarchical policies where high-level latent planners propose grasp candidates, while low-level Cartesian controllers handle fine adjustments via operational space control.

5.4.3 Reinforcement Learning (RL)

Reinforcement Learning frames grasp execution as an optimal control problem within a Markov Decision Process (MDP). Here, the agent interacts with the environment by choosing actions u_t that evolve the state x_t of the robot hand–object system. Over many episodes, it aims to maximize the cumulative reward, such as grasp success, object stability, and energy efficiency.

For MDPs with finite horizons, the objective is as follows:

$$\max_{\pi} \mathbb{E}\left[R(x_T) + \sum_{t=0}^{T-1} r(x_t, u_t)\right],$$

where T is a terminal time (e.g., once the object is grasped). In contrast, infinite-horizon formulations maximize a discounted sum:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t)\right].$$

where (γ) is the discount factor.

Realistic grasp scenarios often involve uncertainty, so the robot receives noisy observations z_t. The agent may maintain a belief state $b_t(x)$ that updates via Bayesian filtering. POMDP formulations are more computationally demanding but enable robust policies that handle sensor noise and missing information.

5.4.4 Model-Free Reinforcement Learning

Pure reinforcement learning, without reliance on expert demonstrations, has become an increasingly viable approach for dexterous manipulation. These methods often leverage Proximal Policy Optimization (PPO) and benefit from structured action spaces such as eigengrasps derived from human hand kinematics to reduce the complexity of controlling high-DOF hands while preserving flexibility.

A prominent strategy across these methods is the integration of affordance-guided exploration, where semantic or geometric cues guide the RL policy toward meaningful grasp regions. Some approaches incorporate pretrained visual encoders or supervised affordance models to identify task-relevant contact areas, especially in tool-use contexts like DexFunc [114],GRAFF [18]. Others avoid semantic labeling entirely,

instead relying on geometric priors like contact point distributions or simulated hand-object interactions to learn robust grasp behaviors like DexPoint[197].

In terms of generalization, these systems vary in their reliance on task specificity versus object diversity. Policies trained on narrow domains (e.g., hammering tasks) tend to show high real-world success rates [114], while those trained across broader object sets aim for robustness to unseen geometries, lighting conditions, or sensor noise [197, 21]. Point cloud representations and domain randomization have proven particularly effective in bridging the sim-to-real gap for these general-purpose models.

Another thread is the emergence of hierarchical policy architectures that enable more complex, dynamic interactions. By decomposing control into low-level grasp correction and high-level manipulation planning, some frameworks can handle full manipulation trajectories rather than just stable grasps such as D-Grasp [125]. While such approaches excel in simulation and enable richer behaviors, they often rely on known object geometry or motion-capture supervision.

Overall, pure RL methods have made impressive strides in achieving real-world dexterity—with some reporting 80–100% task success rates—yet challenges remain in sample efficiency, task generalization, and dynamic interaction. Techniques like curriculum learning, multi-stage reward shaping, and architectural modularity are emerging as promising solutions to these bottlenecks.

5.4.5 Curriculum and Residual Learning

Curriculum learning and residual learning have emerged as complementary strategies for addressing the complexity of dexterous manipulation—particularly the challenges posed by high-DOF control, diverse object geometries, and sim-to-real transfer. Both approaches aim to guide learning by structuring it over time: curriculum methods gradually increase task difficulty, while residual learning incrementally refines base policies without starting from scratch.

A recurring theme in curriculum-based approaches is progressive task or object complexity. Instead of training on the full task distribution from the outset, policies are first exposed to simpler scenarios—e.g., single objects, constrained motion, or static settings—and later fine-tuned on more complex conditions. This idea is reflected in frameworks like UniDexGrasp, which clusters objects by geometry and introduces complexity in stages, and GraspXL [12], which first teaches finger-level precision before tackling dynamic object motion.

Some methods build curricula not just over object types, but over latent geometry representations. UniDexGrasp++ [13] and ResDex [113] use shape-aware embeddings derived from point clouds to organize training, enabling smoother generalization across related geometries. ResDex goes further by employing a residual learning framework based on expert-specialized sub-policies, blending their outputs via a hyper-policy to adapt to new objects. This modularity helps mitigate catastrophic errors and improves generalization beyond what shape clustering alone can achieve.

Recent methods have incorporated Transformer architectures into dexterous manipulation pipelines, often aligning with implicit curriculum learning through staged training or data diversity. In ALOHA Unleashed [15], a Transformer policy is trained via behavior cloning on a large teleoperation dataset, where increasing demonstration complexity effectively structures learning from simple to intricate tasks. Similarly, Object-Centric Dexterous Manipulation [171] uses a Transformer for high-level trajectory generation, paired with a low-level PPO controller. By decoupling motion planning from control and augmenting training with varied object properties and goals, this architecture introduces an implicit curriculum that supports generalization to novel tasks and kinematics.

Beyond fixed curricula, adaptive schemes also play a role. DemoStart [14] uses an auto-curriculum that dynamically adjusts environment difficulty based on policy performance, while also injecting partial demonstration states to bootstrap training. This helps balance exploration and exploitation and accelerates learning in more complex configurations.

Residual learning, when paired with curriculum stages or hierarchical control, enables fine-grained refinement of existing policies. By layering residual components atop a general-purpose policy—often trained on central examples from an object cluster—systems can adjust for shape-specific edge cases without destabilizing prior knowledge [113]. Even methods not explicitly labeled as residual, like SeqDex [172] or DexFunc [114], implicitly adopt this idea: decomposing complex tasks into sub-phases (e.g., pre-grasp adjustment, contact, fine-tuning) and refining each stage either sequentially or through correction modules.

Across these approaches, we see a consistent pattern: structure enables generalization. Curriculum learning provides smoother optimization landscapes, while residual learning injects flexibility into otherwise rigid policies. Together, they enable robust performance across variable object geometries and manipulation tasks, and form a key backbone of recent advances in generalizable, real-world dexterous control.

5.4.6 Teacher–Student Distillation

Teacher–student distillation has become a key strategy for enabling vision-based policies to benefit from the precision and stability of state-based control. In this framework, a teacher policy is trained in simulation using privileged information—such as object kinematics, contact forces, or full state observations—which allows it to converge quickly on effective manipulation strategies. A student policy is then trained to imitate the teacher using realistic sensory inputs like RGB-D images or proprioceptive data, bridging the sim-to-real gap.

Privileged Teacher Policies Modern teacher policies leverage various forms of privileged information to achieve robust simulation performance, with recent advances pushing the boundaries of what these privileged systems can accomplish. The most established approach uses **full-state teachers** that utilize ground-truth object poses and contact forces, as demonstrated by [83] and [198] which achieve $\geq 90\%$ success rates in simulation through complete kinematic awareness. Building on this foundation, **physics-aware teachers** like [177] incorporate analytical contact models through evidential grasp quality estimators, while [181] shows how force-domain diffusion policies can create multi-modal teachers that combine state information with synthetic tactile sensors. The field has seen significant breakthroughs through architectures like [187]'s graph embodiment transformer that enables zero-shot generalization across different hand morphologies, and [200]'s multi-head skill transformers that decompose long-horizon manipulation tasks into executable sub-skills. Further innovations include [201]'s framework for learning task-oriented grasping directly from human videos and [203]'s orientation-aware RL that respects the geometric constraints of dexterous manipulation. These privileged systems collectively form a powerful toolkit for generating training data and behavioral priors that can be distilled into more practical vision-based policies, with each approach offering complementary advantages in terms of simulation fidelity, task generalization, or training efficiency.

Vision-based Student Policies Vision-based student policies bridge the sim-to-real gap by distilling privileged teacher knowledge into systems that operate on raw sensory inputs. The most widely used approach is DAgger-based distillation [17], where student training is augmented with online corrections from the teacher, ensuring robustness to distribution shift. Methods like UniDexGrasp [83] and ResDex [113] demonstrate that DAgger-based vision policy training outperforms standard behavior cloning and adversarial imitation. UniDexGrasp++ [13] further extends this by jointly distilling both policy and value networks, enabling fine-tuning through actor–critic RL and improving sample efficiency.

Transformer-based models have been leveraged to scale this approach. UniGraspTransformer [196] distills

millions of successful RL trajectories—each generated from object-specific policies—into a single universal Transformer policy. The resulting student generalizes across thousands of objects and poses, offering greater scalability and flexibility than earlier models like UniDexGrasp++.

Several approaches combine distillation with curriculum learning, forming a progressive pipeline where the teacher is repeatedly retrained on increasingly complex tasks, and its updated expertise is passed to the student. This ensures that as new challenges are introduced—such as irregular geometries or dynamic environments—the student benefits from the teacher's growing skillset while avoiding catastrophic forgetting. These iterative distillation pipelines improve robustness and stability in long-horizon training.

Distillation also plays a central role in multi-stage or sequential manipulation tasks. In works like SeqDex [172], multiple teacher policies specialize in sub-tasks (e.g., re-grasping, fine orientation), producing near-optimal reference trajectories. The student, trained to imitate these trajectories, learns the key transitions and intermediate states that ensure task continuity—even if it deviates slightly from the teacher's precise motions. This modular guidance improves resilience to compounding errors during long, complex tasks.

Other variations include reconstructing privileged signals indirectly. For example, FunGrasp [19] equips the student with an LSTM to infer contact information from proprioception, while DextrAH-G [129] uses imitation to bypass complex reward engineering. DemoStart [14] leverages a teacher to generate an offline dataset for behavior cloning using a Perceiver–Actor–Critic architecture, efficiently handling visual and proprioceptive data. Finally, UniDexFPM [20] uses diffusion policy distillation to absorb behaviors from a mixture of expert teachers into a single policy, modeling complex action distributions with fewer demonstrations than traditional imitation.

In all these cases, teacher–student distillation not only accelerates learning but also enables real-world deployment by training student policies directly on realistic sensory data. When combined with curriculum learning or modular design, it offers a powerful pathway to scalable and generalizable dexterous manipulation.

5.5 Imitation Learning for Dexterous Manipulation

Dexterous manipulation tasks involve controlling high-degree-of-freedom robotic hands to grasp and manipulate objects in complex environments. Pure reinforcement learning (RL) often struggles in this domain due to sample inefficiency, difficulty exploring large action spaces, and the tendency to produce jerky or unnatural motions when dealing with multi-fingered hands and intricate contact dynamics [105]. Imitation learning (IL) addresses these challenges by leveraging expert demonstration data, thereby bypassing the need for carefully engineered reward functions and reducing excessive exploration.

We model the robotic grasping problem as a Markov Decision Process (MDP) defined by state space $\mathcal{X}$, control space $\mathcal{U}$, and a transition probability distribution

$$p\big(x_{t+1} \;\big|\; x_t,\, u_t\big),$$

which characterizes the likelihood of transitioning to state x_{t+1} given the current state x_t and action u_t. A stationary policy π prescribes a closed-loop mapping

$$u_t \;=\; \pi\big(x_t\big),$$

where $x_t \in \mathcal{X}$ represents the sensed state (e.g., joint angles, tactile readings), and $u_t \in \mathcal{U}$ is the control command (e.g., joint torques or position targets). Unlike RL, where one seeks to maximize cumulative rewards $r_t = R(x_t, u_t)$, in imitation learning we do not have access to an explicit reward function. Instead, we are provided with a dataset of expert trajectories

$$\mathcal{T} \;=\; \{\xi^{(1)},\, \xi^{(2)},\, \ldots,\, \xi^{(D)}\}, \quad \xi^{(i)} \;=\; \big\{(x_0^{(i)},\, u_0^{(i)}),\, (x_1^{(i)},\, u_1^{(i)}),\, \ldots\big\},$$

where each $\xi^{(i)}$ is generated by an expert policy π^*. The aim of IL is to recover a policy $\hat{\pi}$ that closely replicates π^* across the distribution of states encountered during dexterous grasping.

Problem Formulation. Formally, the imitation learning problem can be stated as follows:

[Imitation Learning Problem] Given an MDP $\big(\mathcal{X}, \mathcal{U}, p(\cdot \mid \cdot, \cdot)\big)$ and a set of expert demonstrations $\mathcal{T}$ generated by an unknown policy π^*, identify a policy $\hat{\pi}$ such that for each state x drawn from the relevant distribution, $\hat{\pi}(x)$ approximates the expert action $\pi^*(x)$.

Equivalently, let each demonstration trajectory be a sequence of state–action pairs $\big(x_t, u_t^*\big)$, where $u_t^* = \pi^*(x_t)$. The goal is to find π_θ (parameterized by θ) that minimizes the expected discrepancy between $\pi_\theta(x)$ and u^*. In dexterous tasks, x_t might include proprioceptive feedback (joint angles, velocities) and tactile or visual information, while u_t specifies motor commands for each finger joint. By directly imitating expert trajectories, IL reduces reliance on reward engineering and alleviates the extensive exploration needed by RL—a critical advantage when dealing with high-dimensional action spaces and complex contact interactions.

Advantages for Dexterous Grasping. Expert demonstrations inherently capture smooth coordination of fingers, contact transitions, and force modulation that would be difficult to encode via hand-crafted rewards. As a result, policies learned through IL exhibit more natural and reliable motions when compared to RL-trained controllers that might explore suboptimal or unstable behaviors. Moreover, IL often drastically lowers the sample complexity: since each expert trajectory provides many valid state–action pairs, the robot can learn useful grasps without exhaustively exploring the entire action space.

IL also facilitates generalization: provided that demonstrations cover diverse object geometries and initial conditions, the learned policy can interpolate to novel scenarios in unstructured environments. This capability is especially important for dexterous manipulation, where the variability in object shape, friction properties, and available grasp surfaces is high.

Approaches to Imitation Learning. Broadly, IL methods fall into two categories: those that directly imitate the expert policy and those that infer the expert's underlying reward function.

Direct Policy Imitation: The simplest form is **behavior cloning**, which treats IL as a supervised learning problem. Given the demonstration pairs (x_t, u_t^*), one trains π_θ to minimize a loss such as mean squared error or cross-entropy between $\pi_\theta(x_t)$ and u_t^*. While straightforward, behavior cloning can suffer from *covariate shift*: small errors compound over time because the policy visits states not covered by the demonstrations. The DAgger (Dataset Aggregation) algorithm mitigates this by iteratively collecting new data: at each iteration, the learner's policy is used to sample states, and the expert is queried for the correct actions, thus expanding the dataset to include states induced by the learner's mistakes.

Inverse Reinforcement Learning (IRL): Instead of learning the policy directly, IRL methods first estimate a reward function R under which the expert trajectories are (approximately) optimal. Once the reward is recovered, standard RL can be applied to find a policy that maximizes R. In IRL, one assumes that the expert's behavior arises from optimizing a latent objective; by recovering this objective, the robot can derive a policy that generalizes more robustly to new states. For dexterous manipulation, IRL can capture subtle preferences—such as favoring grasps that minimize contact forces or avoid slipping—which might be difficult to specify manually.

5.5.1 Behavior Cloning

In the following subsections, we first describe classical direct-imitation algorithms, behavior cloning and DAgger, highlighting their implementation details and typical challenges in dexterous tasks. We then discuss IRL approaches, which focus on inferring expert reward functions to guide subsequent policy optimization.

Behavior Cloning (BC) treats dexterous policy acquisition as supervised learning by directly imitating

expert trajectories. Let the set of expert demonstrations be

$$\Xi \;=\; \big\{\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(D)}\big\}, \quad \xi^{(i)} \;=\; \big\{(x_0^{(i)}, u_0^{*(i)}), (x_1^{(i)}, u_1^{*(i)}), \ldots, (x_{N_i}^{(i)}, u_{N_i}^{*(i)})\big\},$$

where each $x \in \mathcal{X}$ is an observed state of the dexterous hand (e.g., joint angles, tactile readings, visual features) and each $u^* \in \mathcal{U}$ is the expert command (e.g., joint torques or position targets). We seek a parameterized policy $\pi_\theta : \mathcal{X} \to \mathcal{U}$ that mimics the expert by minimizing a suitable discrepancy $L(\,\cdot\,,\,\cdot\,)$ between $\pi_\theta(x)$ and the expert action u. Concretely:

$$\hat{\pi} \;=\; \arg\min_{\pi_\theta} \sum_{\xi \in \Xi} \sum_{(x,\,u^*) \in \xi} L\big(\pi_\theta(x),\, u^*\big)\,,$$

where L can be, for instance, a squared-error loss if we assume a Gaussian policy

$$L\big(\pi_\theta(x), u^*\big) \;=\; \big\|\pi_\theta(x) \;-\; u^*\big\|^2$$

or a negative log-likelihood if $\pi_\theta(\cdot \mid x)$ is modeled as a conditional density. In the Gaussian-policy case, minimizing $\sum L(\pi_\theta(x), u^*)$ is equivalent to maximizing $\sum \log \pi_\theta(u^* \mid x)$. Either way, BC reduces to solving

$$\hat{\pi} \;=\; \arg\min_{\theta} \sum_{i=1}^{D} \sum_{t=0}^{N_i} \big\|\pi_\theta\big(x_t^{(i)}\big) \;-\; u_t^{*(i)}\big\|^2,$$

which, for dexterous manipulation, corresponds to regressing the expert's joint-space commands (or end-effector targets) directly from high-dimensional observations.

BC's main advantage in dexterous grasping is that it bypasses manual reward design and extensive exploration, which are particularly burdensome in high-degree-of-freedom settings [105]. Expert demonstrations implicitly capture smooth finger coordination, appropriate contact transitions, and force modulation. By regressing on these demonstrations, BC often produces natural grasp motions for primitives like pick-and-place, pushing, and simple in-hand adjustments [210, 32]. Additionally, each trajectory in $\mathcal{T}$ can contain hundreds or thousands of (x, u^*) samples, enabling the policy to learn effective behaviors with far fewer environment interactions than pure RL.

Despite its simplicity, BC suffers from distribution shift: at test time, the policy may visit states not represented in $\mathcal{T}$. If π_θ makes a small mistake, it can drift into an out-of-distribution state, compounding errors over time and leading to task failure [121]. This issue is exacerbated in dexterous manipulation because successive finger configurations are strongly correlated and the action space is high dimensional [207]. Consequently, even minor deviations can cause the robot to miss contacts or apply incorrect forces.

Several extensions mitigate BC's limitations. One approach uses hierarchical or segment-based representations to decompose complex tasks into shorter motion primitives. By training subpolicies on these primitives, the decision horizon shrinks, reducing error accumulation [207, 208]. Sequence-modeling techniques such as Transformers predict short action sequences conditioned on recent state histories, improving robustness to drift compared to single-step predictions [208]. To capture multi-modal expert behaviors, conditional energy-based models and mixture-of-Gaussians formulations allow π_θ to represent diverse action modes in ambiguous or cluttered scenarios [204, 205]. More recently, diffusion-based models have been applied to model the full joint distribution over state–action trajectories, enabling the policy to generate coherent sequences even under significant uncertainty [209].

Interactive data-collection schemes like DAgger (Dataset Aggregation) further address covariate shift. In DAgger, one iteratively collects new samples by rolling out π_θ, querying the expert π^* for the correct action at each visited state, and aggregating these annotations into the training set $\mathcal{T}$. This procedure ensures that $\mathcal{T}$ eventually covers states induced by the learned policy, reducing compounding errors in long-horizon dexterous tasks [17].

When expert data originates from human demonstrations—via motion capture or teleoperation—retargeting is crucial due to kinematic differences between human hands and robotic grippers. Retargeting algorithms

map human joint angles, end-effector positions, or contact points to the robot's kinematic structure, often by optimizing to preserve functional intent. For example, VideoDex [135] extracts 3D hand-object interactions from internet videos and retargets those motions to a multifingered robot. Other methods combine optimization-based retargeting with learned inverse kinematics to maintain key grasp contacts despite dissimilar joint limits and hand geometry [199, 150]. Accurate retargeting remains an open challenge, as small mapping errors can significantly reduce grasp stability and success.

Learning from Human Videos: Recent work increasingly focuses on learning directly from human videos, bypassing the need for teleoperation or complex MoCap setups. Approaches like DexVIP [122] leverage techniques to infer 3D hand-object interactions and contact states from 2D video, distilling this information into policies for robot execution. This line of research holds promise for scaling up data collection but faces challenges in accurately estimating 3D geometry, contact forces, and handling occlusions from video data alone.

5.5.2 Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) seeks to recover a reward function $R(s, a)$ that explains a collection of expert demonstrations

$$\mathcal{T} \;=\; \big\{\xi^{(1)},\, \xi^{(2)},\, \ldots,\, \xi^{(D)}\big\}, \quad \xi^{(i)} \;=\; \big\{(x_0^{(i)},\, u_0^{*(i)}),\, (x_1^{(i)},\, u_1^{*(i)}),\, \ldots,\, (x_{N_i}^{(i)},\, u_{N_i}^{*(i)})\big\},$$

. Operating within a finite MDP $\big(\mathcal{X},\, \mathcal{U},\, p(\cdot \mid \cdot, \cdot)\big)$, IRL commonly parameterizes the reward as

$$R(x, u) = w^\top \phi(x, u),$$

with $\phi(x, u)$ a feature vector and w a weight vector to be learned. Recall from the RL section for MDPs with finite horizons, the objective is as follows:

$$\max_{\pi} \mathbb{E}\left[R(x_T) + \sum_{t=0}^{T-1} r(x_t, u_t)\right],$$

Substituting the reward here yields

$$\max_{\pi}\; w^\top \underbrace{\mathbb{E}_\pi\Big[\; \phi(x_T, u_T) \;+\; \sum_{t=0}^{T-1} \phi\big(x_t,\, u_t\big)\Big]}_{\mu_{\mathrm{FH}}(\pi)}.$$

IRL then enforces that the expert's feature expectation score exceeds any other policy's:

$$w^\top \widehat{\mu}_{\mathrm{FH}}\big(\pi^*\big) \;\geq\; \max_{\pi}\; w^\top \mu_{\mathrm{FH}}(\pi) \;-\; \varepsilon,$$

for some slack ε. Once $\hat{w}$ is recovered, we obtain

$$\hat{r}(x, u) = \hat{w}^\top \phi(x, u), \qquad \hat{R}(x_T) = \hat{w}^\top \phi(x_T, u_T),$$

and can re-solve the forward RL problem $\max_\pi \mathbb{E}[\, \hat{R}(x_T) + \sum_{t=0}^{T-1} \hat{r}(x_t, u_t)\,]$ to recover a dexterous policy that reproduces the expert's grasping behavior.

By inferring R from $\mathcal{T}$, IRL is particularly valuable in dexterous manipulation, where manually designing a reward function is often impractical. Robots can thus learn to reproduce complex grasping behaviors that align with expert intent.

Reward normalization and feature masking helps mitigate bias toward demonstrated actions, enhancing IRL stability in high-dimensional grasp spaces. Generative Causal Imitation Learning [8] improved sample

efficiency by combining maximum-entropy IRL with adaptive sampling and neural network approximators for cost functions. ErrP-IRL [3] incorporated error-related potentials to weight demonstrations according to user cognitive feedback, refining a reward function represented by radial basis functions. GraphIRL [138] extracted task-specific graph embeddings from unstructured video demonstrations, enabling cross-domain reward learning without explicit environment correspondence. Naranjo-Campos et al. [4] integrated IRL with Proximal Policy Optimization by introducing a reverse discount factor near goal states to mitigate feature vanishing and boost policy robustness. Finally, Visual IRL [206] extended IRL to human–robot collaboration by using adversarial IRL on demonstration videos and a neuro-symbolic mapping that translates human kinematics into robot joint targets, preserving natural motion dynamics.

IRL enables dexterous manipulators to infer nuanced reward structures directly from expert data, circumventing the need for manual reward specification. This capability allows robots to generalize complex behaviors and adapt to diverse scenarios. Nonetheless, accurately estimating R remains challenging in high-dimensional or sparse-feedback settings, and IRL often requires large volumes of expert demonstrations, which are costly to collect. Future work must focus on improving sample efficiency, handling noisy or sparse demonstrations, and reducing dependence on extensive expert data to fully leverage IRL's potential in dexterous grasping.

5.6 Policy Refinements & Extensions

5.6.1 Reward Engineering & Auxiliary Losses

Reward functions play a pivotal role in training policies for dexterous manipulation. They guide agent behavior across phases—approach, grasp, and object manipulation—while ensuring stability, contact quality, and successful task execution.

Modern manipulation systems often decompose the total reward into motion and grasp objectives:

$$r = r_{\text{goal}} + r_{\text{grasp}}$$

Each term aggregates several shaped sub-rewards as shown in Table 4.

Reward Type	Purpose	Example Formulation
Approach Reward	Move hand toward pre-grasp pose	$r_{\text{approach}} = -\|\hat{p}_{\text{pre}} - p_{\text{hand}}\|_2$, if > 0.2
Pre-grasp Imitation	Align hand pose with demonstrated pre-grasp	$\|p - \hat{p}\| + \|q - \hat{q}\|$
Trajectory Following	Track object's desired motion after grasp	$r_{\text{obj}} = -\|x_{\text{obj}} - x_{\text{target}}\|$, $t \geq \lambda$
Contact Reward	Ensure multi-finger contact with object	$r_{\text{contact}} = \mathbb{1}$[thumb+2 fingers contact]
Force Reward	Encourage appropriate contact force distribution	Weighted sum of forces on graspable vs. non-graspable surfaces
Lift Reward	Promote elevation of grasped object	$r_{\text{lift}} = \|z_{\text{obj}} - z_{\text{goal}}\|$
Anatomical & Regularization	Maintain natural joint limits and smooth motion	Penalty on joint violations and high velocities
Success Reward	Bonus for task completion	$r_{\text{success}} = \mathbb{1}$[goal reached] $\times\, b$

Table 4: Common Reward Components in Dexterous Manipulation

Stable grasping is essential in dexterous manipulation. Rewards often include contact count terms, encouraging multi-finger contact, especially involving the thumb to promote caging. Some methods reward the direction of contact forces if they point inward toward the object.

Dense rewards offer continuous feedback and speed up early training through distance penalties or pose imitation. However, they may lead to overfitting or unintended behaviors. Sparse rewards, like binary task success, better reflect true objectives but suffer from weak learning signals. A common strategy is to start

with dense shaping and shift toward sparse rewards for better generalization.

Reward shaping guides the agent through complex tasks and encourages smooth, natural motion. Terms like pose imitation and joint regularization help during early learning. Still, overly specific shaping can limit exploration or lead to exploitation of reward terms. Effective designs balance guidance with flexibility, as seen in staged frameworks like ResDex [113] and SeqDex [172]. Curriculum learning eases training by gradually increasing difficulty—e.g., delaying object tracking rewards until grasping is achieved. LfD strategies like behavior cloning or pose imitation provide early supervision. Stage-wise approaches (e.g., ResDex [113]) remove imitation terms over time, allowing the policy to focus on task completion. This combination boosts learning efficiency and robustness.

6 Evaluation

We validate a simple heuristic grasp policy at scale using NVIDIA IsaacLab with multi-instance GPU physics. We report three modalities: physics-based *Simulation*, *Mesh Plausibility*, and *Intention* semantics. The objective is to measure (i) physical feasibility and lift robustness under gravity and (ii) consistency of the executed grasp relative to the planned grasp.

Datasets YCB objects only. Objects are provided as USD meshes in `isaaclab_assets/multigrasp_objs_sdf` and grasp candidates are drawn from `example.pkl`. We evaluate three YCB-oriented dataset subsets: GraspXL (ECCV 2024), MultiDex v1.0 (CVPR 2023), and DexFuncGrasp (AAAI 2024). YCB is chosen for standardization across simulators, consistent mesh quality, and reproducibility.

6.1 Heuristic Policy Pipeline

Each simulation follows a fixed sequence:

1. **Flat Approach:** Move to a pre-grasp above the object with fingers fully open.
2. **Cover:** Align palm normal to the object's top surface and descend until proximity/contact triggers.
3. **Pre-Grasp Closure:** Light finger closure to establish coverage without force enforcement.
4. **Full Grasp:** Joint-space closure targeting force-closure while maintaining contact stability.
5. **Gravity & Lift:** Enable gravity, settle for $\approx 0.5\,\mathrm{s}$, then lift vertically by $0.2\,\mathrm{m}$.

6.2 Modalities and Outcomes

- *Simulation*: tests physical success via a height-based criterion.
- *Mesh plausibility*: rejects non-physical interpenetrations and insufficient contacts.
- *Intention*: assesses semantic/task alignment of outcomes.

6.3 Failure Modes

The top three failure modes are thin-lip rims on YCB cups, where micro-slip at convex rims causes borderline height failures; asymmetric YCB tools, where off-axis torques on pliers and knives destabilize pinch contacts during lift; and low-albedo or specular objects, where rendering artifacts reduce VLM visual consistency and lower the intention pass rate.

Table 5: Evaluator outcomes over YCB-only evaluation. Simulation success uses the height criterion; if max_height_achieved − $z_{\text{table}} \geq 0.25\,\text{m}$ at any timestep.

Evaluator	**Modality**	**Primary metrics**	**Pass condition(s)**	**Pass (%)**	**Mean latency**
Simulation (comprehensive)	Physics	success rate; max height; slip rate	height ≥ 0.25 m	58.70	86 ms/grasp
Mesh Plausibility (SDF)	Geometry	penetration volume; contact area; #contacts	penetration $\leq 1 \times 10^{-5}$; contacts ≥ 3	72.40	19 ms/grasp
Intention (VLM)	Semantics	alignment; visual consistency; task score (0–1)	each score ≥ 0.70	69.60	2.3 s/grasp

Table 6: Simulation outcomes on YCB-only subsets. Success: max_height_achieved − $z_{\text{table}} \geq 0.25\,\text{m}$. Slip: post-lift COM velocity spike > 0.2 m/s with drop > 0.15 m.

Dataset (subset)	**Hand(s)**	**YCB objs**	**Success (%)**	**Med. height (m)**	**Slip (%)**
GraspXL (ECCV 2024)	Allegro, LEAP	200.00	51.00	0.35	8.20
MultiDex v1.0 (CVPR 2023)	Allegro, LEAP	29.00	48.00	0.33	9.00
DexFuncGrasp (AAAI 2024)	Allegro, LEAP	280.00	44.00	0.32	9.70

Table 7: Metric definitions and pass conditions.

Metric	**Definition**	**Pass condition**	**Units**
Sim success	Object lifted relative to table height	$\max_t z_{\text{obj}}(t) - z_{\text{table}} \geq 0.25$	m
Slip rate	Post-lift instability event	COM velocity spike > 0.2 with drop > 0.15	m/s; m
Penetration volume	Integrated negative SDF inside mesh	$\leq 1 \times 10^{-5}$	m^3
Contact area	Union of fingertip–object contact patches	report only; used by mesh check	cm^2
#Contacts	Distinct fingertip contacts	≥ 3	count
Intention scores	VLM alignment, visual consistency, task proxy (each in $[0, 1]$)	each ≥ 0.70	unitless

6.4 Takeaways

Physics is the primary gate: candidates pass only if the lift reaches the height threshold. Mesh plausibility then filters interpenetrations and inadequate contact geometry at low cost, and intention scoring enforces task alignment on the survivors. The composite score **0.643** indicates balanced physical, geometric, and semantic quality over YCB objects. To raise it, focus on (i) rim-contact control for thin-lip cups via approach-angle tuning, squeeze/roll trajectories, and compliance or friction estimation; (ii) torque-aware grasping for asymmetric tools using wrist alignment, symmetry penalties, and planned regrasp steps; and (iii) robustness on low-albedo/specular items through improved rendering and sensing (exposure control, HDR or polarization, material-aware domain randomization, and depth/normal channels) to stabilize VLM judgments.

7 Future Research

While recent strides in robotic grasp generation have advanced contact modeling, language conditioning, and generative frameworks, significant challenges persist, particularly in generalizing across diverse and deformable objects; addressing this necessitates exploring unsupervised learning and fine-grained control methods to extrapolate grasp configurations for novel items. Bridging the gap between controlled lab settings and dynamic real-world scenarios requires integrating richer multimodal data, including tactile feedback and refinement techniques. Concurrently, the computational demands of generative models hinder real-time deployment, driving research towards more efficient architectures to improve speed and scalability for resource-constrained applications. Further advancements lie in enhancing semantic understanding by developing adaptive LLMs beyond the static interpretations in models, enabling nuanced task execution, and integrating human-robot collaboration mechanisms utilizing intuitive interfaces for real-time adjustments. Finally, ethical considerations and safety must be seriously considered. Incorporating fail-safe mechanisms, ensuring transparency, establishing standardized safety benchmarks, and developing richer, dynamic datasets are necessary to foster trust and enable robust, adaptable grasping systems for human-centric environments.

References

[1] A. Bicchi, V. Kumar, *Proceedings of IEEE ICRA* **2000**.

[2] J. Jian, X. Liu, Z. Chen, M. Li, J. Liu, R. Hu, G-dexgrasp: Generalizable dexterous grasping synthesis via part-aware prior retrieval and prior-assisted generation, **2025**, URL https://arxiv.org/abs/2503.19457.

[3] P. Kamalaruban, R. Devidze, V. Cevher, A. Singla, Interactive teaching algorithms for inverse reinforcement learning, **2019**, URL https://arxiv.org/abs/1905.11867.

[4] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, Y. Wu, The surprising effectiveness of ppo in cooperative, multi-agent games, **2022**, URL https://arxiv.org/abs/2103.01955.

[5] Y. Zhong, X. Huang, R. Li, C. Zhang, Y. Liang, Y. Yang, Y. Chen, Dexgraspvla: A vision-language-action framework towards general dexterous grasping, **2025**, URL https://arxiv.org/abs/2502.20900.

[6] Y. Zhong, Q. Jiang, J. Yu, Y. Ma, Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness, **2025**, URL https://arxiv.org/abs/2503.08257.

[7] L. Shi, Y. Liu, L. Zeng, B. Ai, Z. Hong, H. Su, Learning adaptive dexterous grasping from single demonstrations, **2025**, URL https://arxiv.org/abs/2503.20208.

[8] G. Swamy, S. Choudhury, J. A. Bagnell, Z. S. Wu, Causal imitation learning under temporally correlated noise, **2022**, URL https://arxiv.org/abs/2202.01312.

[9] L. Pinto, A. Gupta, Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours, **2015**, URL https://arxiv.org/abs/1509.06825.

[10] M. Baghbahari, A. Behal, Grasping using tactile sensing and deep calibration, **2019**, URL https://arxiv.org/abs/1907.09656.

[11] A. SaLoutos, H. Kim, E. Stanger-Jones, M. Guo, S. Kim, Towards robust autonomous grasping with reflexes using high-bandwidth sensing and actuation, **2023**, URL https://arxiv.org/abs/2209.11367.

[12] H. Zhang, S. Christen, Z. Fan, O. Hilliges, J. Song, In *European Conference on Computer Vision (ECCV)*. **2024** .

[13] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, H. Wang, *arXiv preprint arXiv:2304.00464* **2023**.

[14] M. Bauza, J. E. Chen, V. Dalibard, N. Gileadi, R. Hafner, M. F. Martins, J. Moore, R. Pevceviciute, A. Laurens, D. Rao, M. Zambelli, M. Riedmiller, J. Scholz, K. Bousmalis, F. Nori, N. Heess, Demostart: Demonstration-led auto-curriculum applied to sim-to-real with multi-fingered robots, **2024**, URL https://arxiv.org/abs/2409.06613.

[15] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, A. Wahid, In P. Agrawal, O. Kroemer, W. Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*. PMLR, **2025** 1910–1924, URL https://proceedings.mlr.press/v270/zhao25b.html.

[16] J. Hang, X. Lin, T. Zhu, X. Li, R. Wu, X. Ma, Y. Sun, In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38. **2024** 10306–10313.

[17] S. Ross, G. J. Gordon, J. A. Bagnell, A reduction of imitation learning and structured prediction to no-regret online learning, **2011**, URL https://arxiv.org/abs/1011.0686.

[18] P. Mandikal, K. Grauman, In *IEEE International Conference on Robotics and Automation (ICRA)*. **2021** .

[19] L. Huang, H. Zhang, Z. Wu, S. Christen, J. Song, *IEEE Robotics and Automation Letters* **2025**.

[20] T. Wu, Y. Gan, M. Wu, J. Cheng, Y. Yang, Y. Zhu, H. Dong, Dexterous functional pre-grasp manipulation with diffusion policy, **2024**, URL `https://arxiv.org/abs/2403.12421`.

[21] C. Bao, H. Xu, Y. Qin, X. Wang, Dexart: Benchmarking generalizable dexterous manipulation with articulated objects, **2023**, URL `https://arxiv.org/abs/2305.05706`.

[22] J. Romero, D. Tzionas, M. J. Black, *ACM Transactions on Graphics* **2017**, *36*, 6 1–17.

[23] A. Agarwal, S. Uppal, K. Shaw, D. Pathak, *arXiv preprint arXiv:2312.02975* **2023**.

[24] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, L. Shao, $\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping, **2025**, URL `https://arxiv.org/abs/2410.01702`.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, **2021**, URL `https://arxiv.org/abs/2106.09685`.

[26] E. Corona, A. Pumarola, G. Alenyà, F. Moreno-Noguer, G. Rogez, In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2020** 5030–5040.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, **2017**, URL `https://arxiv.org/abs/1707.06347`.

[28] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, **2018**, URL `https://arxiv.org/abs/1801.01290`.

[29] Y. Liu, Y. Yang, Y. Wang, X. Wu, J. Wang, Y. Yao, S. Schwertfeger, S. Yang, W. Wang, J. Yu, X. He, Y. Ma, In K. Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. International Joint Conferences on Artificial Intelligence Organization, **2024** 6859–6867, URL `https://doi.org/10.24963/ijcai.2024/758`, Main Track.

[30] M. Qi, Z. Zhao, H. Ma, Human grasp generation for rigid and deformable objects with decomposed vq-vae, **2025**, URL `https://arxiv.org/abs/2501.05483`.

[31] A. van den Oord, O. Vinyals, K. Kavukcuoglu, Neural discrete representation learning, **2018**, URL `https://arxiv.org/abs/1711.00937`.

[32] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al., *The International Journal of Robotics Research* **2020**, *39*, 1 3.

[33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, *arXiv preprint arXiv:1509.02971* **2015**.

[34] I. Popov, N. Heess, T. Lillicrap, R. Hafner, G. Barth-Maron, M. Vecerik, T. Lampe, Y. Tassa, T. Erez, M. Riedmiller, *arXiv preprint arXiv:1704.03073* **2017**.

[35] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, T. Funkhouser, In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, **2018** 4238–4245.

[36] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, V. Kumar, In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, **2019** 3651–3657.

[37] A. Gupta, C. Eppner, S. Levine, P. Abbeel, In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, **2016** 3786–3793.

[38] V. Kumar, E. Todorov, S. Levine, In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2016** 378–383.

[39] A. Nagabandi, K. Konolige, S. Levine, V. Kumar, In *Conference on robot learning*. PMLR, **2020** 1101–1112.

[40] M. Omer, R. Ahmed, B. Rosman, S. F. Babikir, In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE, **2021** 1–6.

[41] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, A. Garg, In *European Conference on Computer Vision*. **2022** 201–221.

[42] D. Turpin, T. Zhong, S. Zhang, G. Zhu, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, A. Garg, In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2023** 8082–8089.

[43] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, S. Huang, In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2023** 8068–8074.

[44] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, H. Wang, In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2023** 11359–11366.

[45] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, H. Wang, In *8th Annual Conference on Robot Learning*. **2024** .

[46] J. Chen, Y. Chen, J. Zhang, H. Wang, In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, **2024** 5281–5288.

[47] M. Liu, Z. Pan, K. Xu, K. Ganguly, D. Manocha, *arXiv preprint arXiv:2002.01530* **2020**.

[48] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, S.-C. Zhu, *IEEE Robotics and Automation Letters* **2021**, *7*, 1 470.

[49] W. Xu, J. Zhang, T. Tang, Z. Yu, Y. Li, C. Lu, *IEEE Robotics and Automation Letters* **2024**.

[50] L. Zhang, K. Bai, G. Huang, Z. Bing, Z. Chen, A. Knoll, J. Zhang, *arXiv preprint arXiv:2404.08844* **2024**.

[51] Y. Li, B. Liu, Y. Geng, P. Li, Y. Yang, Y. Zhu, T. Liu, S. Huang, *IEEE Robotics and Automation Letters* **2024**.

[52] A. Miller, P. Allen, *IEEE Robotics Automation Magazine* **2004**, *11*, 4 110.

[53] M. Ciocarlie, C. Goldfeder, P. Allen, In *Robotics: Science and systems manipulation workshop-sensing and adapting to the real world*. **2007** .

[54] C. Goldfeder, M. Ciocarlie, H. Dang, P. K. Allen, In *2009 IEEE International Conference on Robotics and Automation*. **2009** 1710–1716.

[55] J. Lundell, F. Verdoja, V. Kyrki, *IEEE Robotics and Automation Letters* **2021**, *6*, 4 6899.

[56] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, C. Schmid, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2019** 11807–11816.

[57] L. F. Casas, N. Khargonkar, B. Prabhakaran, Y. Xiang, In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, **2024** 2978–2984.

[58] Y. Li, W. Wei, D. Li, P. Wang, W. Li, J. Zhong, In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, **2022** 714–720.

[59] Y. Shao, C. Xiao, *IEEE Robotics and Automation Letters* **2024**.

[60] W. Cho, J. Lee, M. Yi, M. Kim, T. Woo, D. Kim, T. Ha, H. Lee, J.-H. Ryu, W. Woo, et al., In *European Conference on Computer Vision*. Springer, **2024** 284–303.

[61] L. Yang, K. Li, X. Zhan, F. Wu, A. Xu, L. Liu, C. Lu, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2022** 20953–20962.

[62] X. Zhan, L. Yang, Y. Zhao, K. Mao, H. Xu, Z. Lin, K. Li, C. Lu, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **2024** 445–456.

[63] O. Taheri, N. Ghorbani, M. J. Black, D. Tzionas, In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, **2020** 581–600.

[64] J. Jian, X. Liu, M. Li, R. Hu, J. Liu, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. **2023** 14713–14724.

[65] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, O. Hilliges, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **2023** 12943–12954.

[66] S. Hampali, M. Rad, M. Oberweger, V. Lepetit, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2020** 3196–3206.

[67] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, et al., In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2021** 9044–9053.

[68] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, J. Hays, In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, **2020** 361–378.

[69] S. Brahmbhatt, C. Ham, C. C. Kemp, J. Hays, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. **2019** 8709–8719.

[70] Y. Liu, H. Yang, X. Si, L. Liu, Z. Li, Y. Zhang, Y. Liu, L. Yi, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **2024** 21740–21751.

[71] G. Garcia-Hernando, S. Yuan, S. Baek, T.-K. Kim, In *Proceedings of the IEEE conference on computer vision and pattern recognition*. **2018** 409–419.

[72] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, M. Pollefeys, In *Proceedings of the IEEE/CVF international conference on computer vision*. **2021** 10138–10148.

[73] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, L. Yi, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **2022** 21013–21022.

[74] T. Zhu, R. Wu, X. Lin, Y. Sun, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. **2021** 15741–15751.

[75] Y. Wang, C. Guo, L. Cheng, H. Jiang, Regiongrasp: A novel task for contact region controllable hand grasp generation, **2024**, URL `https://arxiv.org/abs/2410.07995`.

[76] Z. Zhang, H. Wang, Z. Yu, Y. Cheng, A. Yao, H. J. Chang, Nl2contact: Natural language guided 3d hand-object contact modeling with diffusion model, **2024**, URL `https://arxiv.org/abs/2407.12727`.

[77] P. Grady, C. Tang, C. D. Twigg, M. Vo, S. Brahmbhatt, C. C. Kemp, Contactopt: Optimizing contact to improve grasps, **2021**, URL `https://arxiv.org/abs/2104.07267`.

[78] H. Jiang, S. Liu, J. Wang, X. Wang, Hand-object contact consistency reasoning for human grasps generation, **2021**, URL `https://arxiv.org/abs/2104.03304`.

[79] F. Zhao, D. Tsetserukou, Q. Liu, Graingrasp: Dexterous grasp generation with fine-grained contact guidance, **2024**, URL `https://arxiv.org/abs/2405.09310`.

[80] S. Liu, Y. Zhou, J. Yang, S. Gupta, S. Wang, Contactgen: Generative contact modeling for grasp generation, **2023**, URL https://arxiv.org/abs/2310.03740.

[81] S. Brahmbhatt, A. Handa, J. Hays, D. Fox, In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. **2019** 2386–2393.

[82] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, S. Huang, Gendexgrasp: Generalizable dexterous grasping, **2023**, URL https://arxiv.org/abs/2210.00722.

[83] Y. Xu, W. Wan, J. Zhang, H. Liu, Z. Shan, H. Shen, R. Wang, H. Geng, Y. Weng, J. Chen, T. Liu, L. Yi, H. Wang, Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy, **2023**, URL https://arxiv.org/abs/2303.00938.

[84] Z. Zhang, L. Zhou, C. Liu, Z. Liu, C. Yuan, S. Guo, R. Zhao, M. H. A. Jr., F. E. Tay, Dexgraspdiffusion: Diffusion-based unified functional grasp synthesis method for multi-dexterous robotic hands, **2024**, URL https://arxiv.org/abs/2407.09899.

[85] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, S.-C. Zhu, Diffusion-based generation, optimization, and planning in 3d scenes, **2023**, URL https://arxiv.org/abs/2301.06015.

[86] Z. Weng, H. Lu, D. Kragic, J. Lundell, Dexdiffuser: Generating dexterous grasps with diffusion models, **2024**, URL https://arxiv.org/abs/2402.02989.

[87] J. Zhang, W. Xu, Z. Yu, P. Xie, T. Tang, C. Lu, *IEEE Robotics and Automation Letters* **2025**, *10*, 2 995.

[88] X. Chang, Y. Sun, Text2grasp: Grasp synthesis by text prompts of object grasping parts, **2024**, URL https://arxiv.org/abs/2404.15189.

[89] H. Li, W. Mao, W. Deng, C. Meng, H. Fan, T. Wang, P. Tan, H. Wang, X. Deng, Multi-graspllm: A multimodal llm for multi-hand semantic guided grasp generation, **2025**, URL https://arxiv.org/abs/2412.08468.

[90] Y.-L. Wei, J.-J. Jiang, C. Xing, X.-T. Tan, X.-M. Wu, H. Li, M. Cutkosky, W.-S. Zheng, Grasp as you say: Language-guided dexterous grasp generation, **2024**, URL https://arxiv.org/abs/2405.19291.

[91] K. Li, J. Wang, L. Yang, C. Lu, B. Dai, Semgrasp: Semantic grasp generation via language aligned discretization, **2024**, URL https://arxiv.org/abs/2404.03590.

[92] Y. Ye, A. Gupta, K. Kitani, S. Tulsiani, G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis, **2024**, URL https://arxiv.org/abs/2404.12383.

[93] K. Karunratanakul, J. Yang, Y. Zhang, M. Black, K. Muandet, S. Tang, Grasping field: Learning implicit representations for human grasps, **2020**, URL https://arxiv.org/abs/2008.04451.

[94] Y. Zhang, J. Hang, T. Zhu, X. Lin, R. Wu, W. Peng, D. Tian, Y. Sun, *IEEE Robotics and Automation Letters* **2023**, *8*, 5 3094.

[95] S. Christen, S. Hampali, F. Sener, E. Remelli, T. Hodan, E. Sauser, S. Ma, B. Tekin, In *SIGGRAPH Asia 2024 Conference Papers*. ACM, **2024** 1–11, URL http://dx.doi.org/10.1145/3680528.3687563.

[96] Y. Zhong, Q. Jiang, J. Yu, Y. Ma, *arXiv preprint arXiv:2503.08257* **2025**.

[97] Y. Zhu, Y. Zhong, Z. Yang, P. Cong, J. Yu, X. Zhu, Y. Ma, *arXiv preprint arXiv:2503.14329* **2025**.

[98] J. Lu, H. Kang, H. Li, B. Liu, Y. Yang, Q. Huang, G. Hua, Ugg: Unified generative grasping, **2024**, URL https://arxiv.org/abs/2311.16917.

[99] P. Li, Z. Wang, M. Liu, H. Liu, C. Chen, In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24. ACM, **2024** 273–281, URL http://dx.doi.org/10.1145/3664647.3680597.

[100] Y.-L. Wei, M. Lin, Y. Lin, J.-J. Jiang, X.-M. Wu, L.-A. Zeng, W.-S. Zheng, Afforddexgrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance, **2025**, URL https://arxiv.org/abs/2503.07360.

[101] Z. Zhao, M. Qi, H. Ma, Decomposed vector-quantized variational autoencoder for human grasp generation, **2024**, URL https://arxiv.org/abs/2407.14062.

[102] G.-H. Xu, Y.-L. Wei, D. Zheng, X.-M. Wu, W.-S. Zheng, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **2024** 17933–17942.

[103] T. Zhong, C. Allen-Blanchette, *arXiv preprint arXiv:2503.04123* **2025**.

[104] T. Zhu, R. Wu, J. Hang, X. Lin, Y. Sun, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2023**, *45*, 10 12521.

[105] I. Radosavovic, X. Wang, L. Pinto, J. Malik, In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. **2021** 7865–7871.

[106] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, X. Wang, In *European Conference on Computer Vision*. Springer, **2022** 570–587.

[107] Y.-H. Wu, J. Wang, X. Wang, In *Conference on Robot Learning*. PMLR, **2023** 618–629.

[108] A. Mousavian, C. Eppner, D. Fox, 6-dof graspnet: Variational grasp generation for object manipulation, **2019**, URL https://arxiv.org/abs/1905.10520.

[109] M. Sundermeyer, A. Mousavian, R. Triebel, D. Fox, Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes, **2021**, URL https://arxiv.org/abs/2103.14127.

[110] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. J. Kohlhoff, T. Kröger, J. J. Kuffner, K. Goldberg, *2016 IEEE International Conference on Robotics and Automation (ICRA)* **2016**, 1957–1964.

[111] R. Bellman, *Journal of mathematics and mechanics* **1957**, 679–684.

[112] N. Khargonkar, N. Song, Z. Xu, B. Prabhakaran, Y. Xiang, Neuralgrasps: Learning implicit representations for grasps of multiple robotic hands, **2022**, URL https://arxiv.org/abs/2207.02959.

[113] Z. Huang, H. Yuan, Y. Fu, Z. Lu, In *The Thirteenth International Conference on Learning Representations*. **2025** URL https://openreview.net/forum?id=BUj9VSCoET.

[114] A. Agarwal, S. Uppal, K. Shaw, D. Pathak, Dexterous functional grasping, **2023**, URL https://arxiv.org/abs/2312.02975.

[115] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin,

S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, Gpt-4 technical report, **2024**, URL `https://arxiv.org/abs/2303.08774`.

[116] S. Chen, J. Bohg, C. K. Liu, Springgrasp: Synthesizing compliant, dexterous grasps under shape uncertainty, **2024**, URL `https://arxiv.org/abs/2404.13532`.

[117] T. H. E. Tse, Z. Zhang, K. I. Kim, A. Leonardis, F. Zheng, H. J. Chang, S^2contact: Graph-based network for 3d hand-object contact estimation with semi-supervised learning, **2023**, URL `https://arxiv.org/abs/2208.00874`.

[118] B. Zuo, Z. Zhao, W. Sun, X. Yuan, Z. Yu, Y. Wang, *IEEE Transactions on Visualization and Computer Graphics* **2024**, 1–13.

[119] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, M. Le, Flow matching for generative modeling, **2023**, URL `https://arxiv.org/abs/2210.02747`.

[120] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, **2020**, URL `https://arxiv.org/abs/2006.11239`.

[121] S. Ross, D. Bagnell, In Y. W. Teh, M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*. PMLR, Chia Laguna Resort, Sardinia, Italy, **2010** 661–668, URL `https://proceedings.mlr.press/v9/ross10a.html`.

[122] Y.-H. Wu, J. Wang, X. Wang, In K. Liu, D. Kulic, J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*. PMLR, **2023** 618–629, URL `https://proceedings.mlr.press/v205/wu23a.html`.

[123] Z. Luo, J. Cao, S. Christen, A. Winkler, K. Kitani, W. Xu, Grasping diverse objects with simulated humanoids, **2024**, URL `https://arxiv.org/abs/2407.11385`.

[124] Z. Chen, S. Chen, E. Arlaud, I. Laptev, C. Schmid, Vividex: Learning vision-based dexterous manipulation from human videos, **2025**, URL `https://arxiv.org/abs/2404.15709`.

[125] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, O. Hilliges, In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2022** 20545–20554.

[126] J. Lu, H. Kang, H. Li, B. Liu, Y. Yang, Q. Huang, G. Hua, *UGG: Unified Generative Grasping*, 414–433, Springer Nature Switzerland, ISBN 9783031728556, **2024**.

[127] W. Wang, F. Wei, L. Zhou, X. Chen, L. Luo, X. Yi, Y. Zhang, Y. Liang, C. Xu, Y. Lu, J. Yang, B. Guo, Unigrasptransformer: Simplified policy distillation for scalable dexterous robotic grasping, **2025**, URL https://arxiv.org/abs/2412.02699.

[128] H. Yuan, B. Zhou, Y. Fu, Z. Lu, Cross-embodiment dexterous grasping with reinforcement learning, **2024**, URL https://arxiv.org/abs/2410.02479.

[129] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, K. V. Wyk, In *8th Annual Conference on Robot Learning*. **2024** URL https://openreview.net/forum?id=S2Jwb0i7HN.

[130] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, J. Malik, Hand-object interaction pretraining from videos, **2024**, URL https://arxiv.org/abs/2409.08273.

[131] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, **2021**, URL https://arxiv.org/abs/2003.05991.

[132] M. T. Ciocarlie, P. K. Allen, *The International Journal of Robotics Research* **2009**, *28*, 7 851.

[133] J. Zhang, M. Li, Y. Feng, C. Yang, *Multimedia Tools and Applications* **2020**, *79* 2427.

[134] V. G. Moudgal, K. M. Passino, S. Yurkovich, *IEEE transactions on control systems technology* **1994**, *2*, 4 392.

[135] K. Shaw, S. Bahl, D. Pathak, Videodex: Learning dexterity from internet videos, **2022**, URL https://arxiv.org/abs/2212.04498.

[136] J. Orbik, A. Agostini, D. Lee, In *2021 IEEE International Conference on Development and Learning (ICDL)*. **2021** 1–7.

[137] I. Batzianoulis, F. Iwane, S. Wei, C. Correia, R. Chavarriaga, J. d. R. Millan, A. Billard, *Communications Biology* **2021**, *4*.

[138] S. Kumar, J. Zamora, N. Hansen, R. Jangir, X. Wang, Graph inverse reinforcement learning from diverse videos, **2022**, URL https://arxiv.org/abs/2207.14299.

[139] F. Naranjo-Campos, J. Victores, C. Balaguer, *Applied Sciences* **2024**, *14* 11131.

[140] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, P. Agrawal, *Science Robotics* **2023**, *8*, 84.

[141] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, S. Levine, Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, **2018**, URL https://arxiv.org/abs/1709.10087.

[142] A. Nair, A. Gupta, M. Dalal, S. Levine, Awac: Accelerating online reinforcement learning with offline datasets, **2021**, URL https://arxiv.org/abs/2006.09359.

[143] S. Dasari, A. Gupta, V. Kumar, Learning dexterous manipulation from exemplar object trajectories and pre-grasps, **2023**, URL https://arxiv.org/abs/2209.11221.

[144] K. Srinivasan, E. Heiden, I. Ng, J. Bohg, A. Garg, In *International Symposium of Robotics Research (ISRR)*. **2024** .

[145] Z. Huang, R. Boulic, N. M. Thalmann, D. Thalmann, In R. Earnshaw, J. Vince, editors, *Computer Graphics*, 235–253. Academic Press, Boston, ISBN 978-0-12-227741-2, **1995**, URL https://www.sciencedirect.com/science/article/pii/B9780122277412500219.

[146] M. Li, H. Yin, K. Tahara, A. Billard, In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2014** 6784–6791.

[147] X. Li, Z. Chen, C. Ma, *Assembly Automation* **2021**, *41*, 2 208.

[148] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, Y. Bengio, Improving and generalizing flow-based generative models with minibatch optimal transport, **2024**, URL https://arxiv.org/abs/2302.00482.

[149] A. Tong, N. Malkin, K. Fatras, L. Atanackovic, Y. Zhang, G. Huguet, G. Wolf, Y. Bengio, Simulation-free schrödinger bridges via score and flow matching, **2024**, URL https://arxiv.org/abs/2307.03672.

[150] A. Sivakumar, K. Shaw, D. Pathak, *arXiv preprint arXiv:2202.10448* **2022**.

[151] N. Khargonkar, L. F. Casas, , B. Prabhakaran, Y. Xiang, **2024** .

[152] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al., *IEEE Robotics and Automation Letters* **2020**, *5*, 3 3838.

[153] X. Wu, T. Liu, C. Li, Y. Ma, Y. Shi, X. He, *arXiv preprint arXiv:2411.14786* **2024**.

[154] C. Ferrari, J. Canny, In *Proceedings 1992 IEEE International Conference on Robotics and Automation*. **1992** 2290–2295 vol.3.

[155] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, M. Chen, Point-e: A system for generating 3d point clouds from complex prompts, **2022**, URL https://arxiv.org/abs/2212.08751.

[156] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, **2018**, URL https://arxiv.org/abs/1706.08500.

[157] J. Chen, Y. Ke, L. Peng, H. Wang, Dexonomy: Synthesizing all dexterous grasp types in a grasp taxonomy, **2025**, URL https://arxiv.org/abs/2504.18829.

[158] I. Lenz, H. Lee, A. Saxena, *The International Journal of Robotics Research (IJRR)* **2015**, *34*, 4-5 705.

[159] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio Ojea, K. Goldberg, In *Robotics: Science and Systems (RSS)*. **2017** URL https://arxiv.org/abs/1703.09312.

[160] L. Pinto, A. Gupta, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2016** 3406–3413, URL https://arxiv.org/abs/1509.06825.

[161] B. Wu, I. Akinola, J. Varley, P. K. Allen, In *Conference on Robot Learning (CoRL)*. **2020** 968–979, URL https://arxiv.org/abs/1909.04787.

[162] A. Raffin, O. Sigaud, J. Kober, A. Albu-Schäffer, J. Silvério, F. Stulp, *arXiv preprint arXiv:2310.05808* **2023**.

[163] D. Morrison, P. Corke, J. Leitner, In *Robotics: Science and Systems (RSS)*. **2018** URL https://arxiv.org/abs/1804.05172.

[164] N. P. Garg, D. Hsu, W. S. Lee, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2019** 2751–2757, URL https://motion.comp.nus.edu.sg/wp-content/uploads/2019/02/icra2019grasping.pdf.

[165] D. Zhao, J. Oh, *IEEE Robotics and Automation Letters* **2020**, *6*, 2 628.

[166] S. Quan, X. Liang, H. Zhu, M. Hirano, Y. Yamakawa, *Sensors* **2022**, *22*, 11.

[167] K. Koyama, M. Shimojo, T. Senoo, M. Ishikawa, *IEEE Robotics and Automation Letters* **2018**, *3*, 4 3224.

[168] K. Hsiao, L. P. Kaelbling, T. Lozano-Perez, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. **2007** 1228–1233, URL https://people.csail.mit.edu/kjhsiao/webpagefiles/papers/contactreactive.pdf.

[169] L. P. Jentoft, Q. Wan, R. D. Howe, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2014** 6394–6399, URL https://biorobotics.harvard.edu/pubs/2014/ref_conf/JentoftICRA2014_ContactRelative.pdf.

[170] E. Páll, O. Brock, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2021** 10059–10065, URL https://ieeexplore.ieee.org/document/9561065.

[171] Y. Chen, C. Wang, Y. Yang, K. Liu, In *8th Annual Conference on Robot Learning*. **2024** .

[172] Y. Chen, C. Wang, L. Fei-Fei, C. K. Liu, *arXiv preprint arXiv:2309.00987* **2023**.

[173] Z. Luo, J. Cao, S. Christen, A. Winkler, K. Kitani, W. Xu, In *Advances in Neural Information Processing Systems (NeurIPS)*. **2024** URL https://arxiv.org/abs/2407.11385.

[174] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, G. Rogez, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2020** .

[175] Q. Feng, D. S. M. Lema, M. Malmir, H. Li, J. Feng, Z. Chen, A. Knoll, Dexgangrasp: Dexterous generative adversarial grasping synthesis for task-oriented manipulation, **2024**, URL https://arxiv.org/abs/2407.17348.

[176] C. J. Ford, H. Li, J. Lloyd, M. G. Catalano, M. Bianchi, E. Psomopoulou, N. F. Lepora, Tactile-driven gentle grasping for human-robot collaborative tasks, **2023**.

[177] Y. Shi, E. Welte, M. Gilles, R. Rayyes, vmf-contact: Uncertainty-aware evidential learning for probabilistic contact-grasp in noisy clutter, **2025**, URL https://arxiv.org/abs/2411.03591.

[178] L. Röstel, D. Winkelbauer, J. Pitz, L. Sievers, B. Bäuml, Composing dextrous grasping and in-hand manipulation via scoring with a reinforcement learning critic, **2025**, URL https://arxiv.org/abs/2505.13253.

[179] G. Stracquadanio, F. Vasile, E. Maiettini, N. Boccardo, L. Natale, Bring your own grasp generator: Leveraging robot grasp generation for prosthetic grasping, **2025**, URL https://arxiv.org/abs/2503.00466.

[180] B. Zhang, I. Andrussow, A. Zell, G. Martius, The role of tactile sensing for learning reach and grasp, **2025**, URL https://arxiv.org/abs/2502.20367.

[181] Y. Wu, Z. Chen, F. Wu, L. Chen, L. Zhang, Z. Bing, A. Swikir, S. Haddadin, A. Knoll, Tacdiffusion: Force-domain diffusion policy for precise tactile manipulation, **2025**, URL https://arxiv.org/abs/2409.11047.

[182] B. Wang, N. Sridhar, C. Feng, M. V. der Merwe, A. Fishman, N. Fazeli, J. J. Park, Thisthat: Language-gesture controlled video generation for robot planning, **2025**, URL https://arxiv.org/abs/2407.05530.

[183] Z. Wang, A. H. Qureshi, Implicit physics-aware policy for dynamic manipulation of rigid objects via soft body tools, **2025**, URL https://arxiv.org/abs/2502.05696.

[184] H. Raei, E. D. Momi, A. Ajoudani, A reinforcement learning approach to non-prehensile manipulation through sliding, **2025**, URL https://arxiv.org/abs/2502.17221.

[185] T. Gao, S. Nasiriany, H. Liu, Q. Yang, Y. Zhu, Prime: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning, **2024**, URL https://arxiv.org/abs/2403.00929.

[186] H. Chen, T. Kiyokawa, W. Wan, K. Harada, Adaptive grasping of moving objects in dense clutter via global-to-local detection and static-to-dynamic planning, **2025**, URL https://arxiv.org/abs/2502.05916.

[187] A. Patel, S. Song, In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. **2025** .

[188] H. Qi, B. Yi, M. Lambeta, Y. Ma, R. Calandra, J. Malik, *arXiv preprint arXiv:2501.05439* **2025**.

[189] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer, London, 2nd edition, **2010**.

[190] M. W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, John Wiley & Sons, Hoboken, NJ, **2006**.

[191] W. Yuan, S. Dong, E. H. Adelson, *Sensors* **2017**, *17*, 12 2762.

[192] A. Church, J. Lloyd, R. Hadsell, N. F. Lepora, *Nature Reviews Electrical Engineering* **2022**, *1*, 1 1.

[193] K.-W. Lee, Y. Qin, X. Wang, S.-C. Lim, *IEEE Robotics and Automation Letters* **2024**, *9*, 12 10772.

[194] B. Zhang, I. Andrussow, A. Zell, G. Martius, The role of tactile sensing for learning reach and grasp, **2025**, URL https://arxiv.org/abs/2502.20367.

[195] Y. Han, K. Yu, R. Batra, N. Boyd, C. Mehta, T. Zhao, Y. She, S. Hutchinson, Y. Zhao, *IEEE/ASME Transactions on Mechatronics* **2025**, *30*, 1 554.

[196] W. Wang, F. Wei, L. Zhou, X. Chen, L. Luo, X. Yi, Y. Zhang, Y. Liang, C. Xu, Y. Lu, J. Yang, B. Guo, In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. **2025** 12199–12208.

[197] Y. Qin, B. Huang, Z.-H. Yin, H. Su, X. Wang, In K. Liu, D. Kulic, J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*. PMLR, **2023** 594–605, URL https://proceedings.mlr.press/v205/qin23a.html.

[198] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, Y. Zhu, In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. **2025** .

[199] Z. Q. Chen, K. VanWyk, Y. Chao, W. Yang, A. Mousavian, A. Gupta, D. Fox, Dextransfer: Real world multi-fingered dexterous grasping with minimal human demonstrations, **2022**, URL https://arxiv.org/abs/2209.14284.

[200] K. Gao, F. Wang, E. Aduh, D. Randle, J. Shi, Must: Multi-head skill transformer for long-horizon dexterous manipulation with skill progress, **2025**, URL https://arxiv.org/abs/2502.02753.

[201] W. Dong, D. Huang, J. Liu, C. Tang, H. Zhang, In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2025** .

[202] A. G. Allievi, G. Neumann, In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2025** .

[203] M. Schuck, J. Brüdigam, S. Hirche, A. Schoellig, In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, **2025** .

[204] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, J. Tompson, In A. Faust, D. Hsu, G. Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*. PMLR, **2022** 158–168, URL https://proceedings.mlr.press/v164/florence22a.html.

[205] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, L. Pinto, In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc., Red Hook, NY, USA, ISBN 9781713871088, **2022** .

[206] E. Asali, P. Doshi, Visual irl for human-like robotic manipulation, **2024**, URL `https://arxiv.org/abs/2412.11360`.

[207] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al., In *Conference on Robot Learning*. PMLR, **2018** 879–893.

[208] X. Chen, H. Peng, D. Wang, H. Lu, H. Hu, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2023** 14572–14581.

[209] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, S. Song, *The International Journal of Robotics Research* **2023**, 02783649241273668.

[210] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, S. Levine, Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation, **2018**, URL `https://arxiv.org/abs/1806.10293`.