

Review

A Review of Language-Guided Dexterous Grasping with Human-in-the-Loop Adaptation

Eugene Jeong  and Soo-Chul Lim * 

Department of Mechanical, Robotics and Energy Engineering, Dongguk University, 30 Pildong-ro 1-gil, Jung-gu, Seoul 04620, Republic of Korea; jeongyj@dgu.ac.kr

* Correspondence: limsc@dongguk.edu

Abstract

Language-guided dexterous grasping aims to translate natural-language instructions and multimodal observations into multi-fingered grasps that are physically admissible, semantically aligned, functionally suitable, user-adaptive, and robust under execution uncertainty. Existing research remains fragmented across language-conditioned manipulation, dexterous grasp generation, human-in-the-loop learning, and simulation-based validation. This structured scoping review synthesizes these areas through a formulation-oriented analytical lens and organizes prior work around the main decisions involved in language-guided dexterous grasping: candidate generation, admissibility verification, multi-objective evaluation, human-feedback-based adaptation, and simulation-based or digital twin-based validation. We compare language, vision–language, and vision–language–action models, optimization-based and generative grasp methods, feedback mechanisms, and validation strategies in terms of their representations, assumptions, data requirements, evaluation settings, and failure modes. The synthesis indicates that no single paradigm currently provides reliable end-to-end language-guided dexterous grasping: semantic reasoning lacks contact-level guarantees, physically grounded generators provide limited task and user conditioning, human feedback introduces scalability costs, and simulation remains vulnerable to model mismatch. Finally, we identify benchmark requirements and research directions for language-conditioned datasets, semantic alignment, functional suitability, tactile integration, low-burden feedback, execution-risk estimation, and calibrated sim-to-real validation.

Keywords: language-guided dexterous grasping; language-conditioned manipulation; dexterous grasp generation; human-in-the-loop learning; functional grasping; semantic alignment; simulation-based validation; sim-to-real transfer; digital twin



Academic Editors: Fajie Wang, Ji Lin and Yingbin Chai

Received: 31 May 2026

Revised: 16 July 2026

Accepted: 22 July 2026

Published: 29 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

MSC: 68T40

1. Introduction

Robotic grasping has long been studied as a fundamental problem in manipulation. Classical work established analytical foundations based on closure properties, grasp matrices, frictional contact models, grasp-wrench spaces, and grasp-quality measures [1–3]. GraspIt! and subsequent surveys further organized grasp synthesis around geometric reasoning, contact modeling, search, and physical evaluation [4–7].

The transition toward data-driven grasping was accelerated by synthetic datasets, learned grasp-quality models, and large-scale robot experience. The Dex-Net series

combined analytic grasp metrics with synthetic observations to learn robust grasping policies [8–10], while large-scale real-world learning demonstrated vision-based hand-eye coordination and reinforcement-learning-based grasping [11,12]. Real-time grasp synthesis, point-cloud-based grasp detection, and large-scale benchmarks further improved the efficiency and coverage of six-degree-of-freedom grasp prediction [13–17]. These developments provide important foundations for perception, proposal generation, and physical evaluation, but most use parallel-jaw grippers or other relatively low-dimensional end effectors.

Dexterous robotic hands introduce a substantially more complex decision space. A dexterous grasp may require the simultaneous determination of the global hand pose, high-dimensional finger-joint configuration, multiple hand–object contacts, and force-, pressure-, or compliance-related variables. Small changes in the palm pose or finger configuration can create or remove contacts, introduce self-collision or object penetration, and change grasp stability. Recent research has addressed dexterous manipulation through large-scale simulated grasp datasets, universal grasp proposal and execution frameworks, point-cloud reinforcement learning, and demonstration-augmented dexterous control [18–22]. The preprint-only Rubik’s Cube study [23] further illustrates the difficulty of robust sim-to-real dexterous manipulation. More recent methods employ transformers, diffusion models, physics-aware generation, unified generative formulations, and cross-hand transfer to produce diverse dexterous grasp candidates [24–31]. Nevertheless, these methods primarily address geometric and physical grasp generation and do not by themselves determine which candidate best satisfies a natural-language instruction, downstream function, or user-dependent requirement.

In parallel, large language models (LLMs), vision–language models (VLMs), and vision–language–action (VLA) models have changed how robotic tasks are specified and generalized. Language-based planning methods have shown that pretrained models can decompose instructions, induce skills, select embodiment-compatible actions, revise plans using feedback, and generate executable robot programs [32–37]. Embodied multi-modal and VLA models connect language and perception more directly to robot state and control [38–40]. Open X-Embodiment, Octo, and OpenVLA extend this direction toward cross-embodiment datasets and generalist robot policies [41–43], while recent models investigate explicit three-dimensional structure, flow matching, diffusion-based bimanual control, spatial representations, and speech-conditioned actions [44–49].

Despite these advances, most general VLA systems remain centered on arm-level manipulation, simple grippers, or action spaces in which contact structure and force regulation are represented only implicitly. Extending them to dexterous hands is therefore not simply a matter of increasing the number of output dimensions. The generated behavior must account for hand-specific joint limits, coordinated finger motion, multiple contact modes, collision, reachability, force transmission, and contact transitions. Semantic plausibility learned from language or demonstrations does not guarantee that a generated hand configuration is physically admissible or robust during execution.

Recent studies have begun to address the interface between language and dexterous grasp generation directly, including direct language-conditioned grasp prediction, semantic multi-hand grasp generation, affordance-mediated grasping, large-scale vision–language–grasp modeling, and hierarchical VLA-based dexterous execution [50–54]. These methods establish language-guided dexterous grasping as an emerging research direction, while also exposing unresolved interfaces among semantic grounding, high-dimensional grasp generation, embodiment-specific admissibility, and real-world execution.

Physical stability alone is insufficient for many language-guided tasks. Prior work on semantic grasping, tool-part affordance detection, affordance segmentation, task-oriented

grasping, and actionability prediction has shown that appropriate contacts depend on object function and intended use [55–59]. The same object may require different contact regions and hand orientations depending on whether it is to be lifted, used, inserted, poured from, stored, or handed to another person. A grasp may therefore be mechanically stable but inconsistent with the instruction, obstruct a task-critical region, or prevent the intended downstream action. Conversely, a semantically preferred region may be too small, occluded, unreachable, or physically unsuitable for the current hand. Language-guided dexterous grasping must consequently distinguish mandatory physical and safety requirements from semantic alignment, functional suitability, user preference, and execution risk.

Human-in-the-loop adaptation is relevant because these requirements cannot always be specified completely in advance. Learning from demonstration provides a general mechanism for transferring human behavior to robots [60], while TAMER and preference-based reinforcement learning show how evaluative feedback and comparisons can shape learned behavior [61–63]. Language corrections can revise task interpretation or provide reusable corrective knowledge [64–66], and deployment-time correction or intervention can support adaptation when simulation-trained policies encounter unfamiliar physical conditions [67–69]. These feedback types are not equivalent: demonstrations, language corrections, preferences, approvals, and interventions modify different components of the grasping system and impose different burdens in terms of supervision time, expertise, consistency, and latency.

Simulation and digital twin environments provide a complementary mechanism for training, testing, and adapting grasping systems. MuJoCo, PyBullet, Isaac Gym, and Isaac Sim support rigid-body dynamics, contact simulation, scalable reinforcement learning, sensor simulation, and synthetic scene construction [70–73]. Manipulation benchmarks provide standardized tasks and observations for policy evaluation [74–80], while domain and dynamics randomization and adaptive sim-to-real methods address uncertainty in visual and physical parameters [81–84]. More recent real-to-sim-to-real and digital twin approaches reconstruct or synchronize simulations from physical observations and use them for policy improvement or pre-execution validation [85–87]. However, a visually accurate simulation may remain inaccurate in contact, friction, compliance, actuation, or sensing, and simulation-derived validation should therefore be treated as an uncertain prediction of physical execution rather than as independent evidence of safety.

Existing reviews provide important foundations for individual components of this problem, including analytic and data-driven grasp synthesis [3,5,6], learning from demonstration and preference-based learning [60,63], human-in-the-loop reinforcement learning [88], and imitation or interactive learning for dexterous manipulation [89,90]. These areas have generally been reviewed separately. Limited attention has been given to how language grounding, high-dimensional dexterous grasp generation, differentiated human feedback, and simulation-based validation interact within a common analytical formulation. In particular, prior reviews do not jointly distinguish grasp candidate generation, physical and safety admissibility, multi-objective selection, feedback-dependent adaptation, and simulation-based or digital twin-based execution-risk estimation.

In Table 1, “Limited” indicates that the topic is not addressed or is mentioned only briefly, without a comparative analysis of relevant studies. “Partial” indicates that the topic is discussed in a dedicated subsection, table, or comparative analysis but is not a principal focus of the review. “Strong” indicates that the topic is a principal focus of the review and is systematically synthesized across multiple studies. These labels describe the extent of topic coverage rather than the quality of the cited review. Existing reviews have generally treated language grounding, high-dimensional dexterous grasp generation, human feedback, and simulation-based validation as separate topics.

Table 1. Comparison of existing review papers related to language-guided dexterous grasping.

Review Category	Representative Literature	Language Grounding	Dexterous Grasping	Human Feedback	Simulation Validation	Remaining Gap
Analytic and data-driven grasp synthesis	Classical grasping and data-driven grasp synthesis surveys [3,5,6]	Limited	Partial	Limited	Partial	Limited treatment of language, feedback, and user-dependent grasp selection.
Learning from demonstration and preference learning	Learning from demonstration and preference-based reinforcement learning surveys [60,63]	Limited	Partial	Strong	Limited	Limited separation of admissibility, semantic alignment, functional suitability, user-dependent preference, and execution risk.
Human-in-the-loop reinforcement learning	Human-in-the-loop reinforcement learning survey [88]	Partial	Limited	Strong	Partial	Limited focus on contact-rich dexterous grasp representation.
Dexterous manipulation and interactive imitation	Dexterous manipulation and interactive imitation learning surveys [89,90]	Limited	Strong	Strong	Partial	Limited treatment of language-conditioned grasp selection.
Embodied grasping and foundation-model-based grasping	Embodied grasping review [91]	Partial	Partial	Partial	Partial	Limited separation of admissibility, semantic alignment, functional suitability, user-dependent preference, and execution risk.
LLM, VLM, and VLA robotics	LLM- and VLA-based robotics reviews [92] [†] , [93]	Strong	Limited	Partial	Partial	Limited focus on dexterous contact admissibility and multi-fingered grasp constraints.
Diffusion-model-based manipulation	Diffusion models for robotic manipulation [94]	Partial	Partial	Limited	Partial	Limited integration of language grounding, human feedback, and digital twin validation.

[†] Available only as a preprint at the time of the final literature search.

The principal contributions of this review are as follows:

1. **Structured review and taxonomy:** We conduct a structured scoping review of language-guided robotic manipulation, dexterous grasp generation, human-in-the-loop adaptation, and simulation-based validation, and classify the selected studies according to their inputs, output representations, physical assumptions, feedback mechanisms, evaluation settings, publication status, and principal limitations.
2. **Common analytical formulation:** We introduce a review-level formulation that separates candidate generation, admissibility verification, objective-level grasp evaluation, and final selection, and relates constrained, Pareto-based, weighted, preference-based, and risk-sensitive selection to different grasp-decision requirements.
3. **Cross-paradigm critical synthesis:** We compare language and vision-language-action models, constrained optimization, probabilistic and generative grasp methods, human-feedback mechanisms, and simulation- or digital twin validation evidence according to their representations, uncertainty assumptions, computational and data requirements, characteristic failure modes, and appropriate roles within a complete system.
4. **Evaluation framework and research agenda:** We identify limitations in current candidate-level, simulation-level, real-robot, functional-suitability, and human-centered evaluations and propose benchmark requirements and actionable research questions concerning language-conditioned grasp datasets, semantic-alignment and functional-suitability evaluation, tactile integration, low-burden human feedback, safety, and calibrated sim-to-real validation.

The remainder of this paper is organized as follows. Section 2 describes the review methodology, including the scope, research questions, search strategy, eligibility criteria,

and classification procedure. Section 3 introduces the common problem formulation. Section 4 reviews language-based task reasoning, vision–language grounding, and vision–language–action policies. Section 5 examines dexterous grasp representations, generation methods, tactile sensing, datasets, and evaluation. Section 6 analyzes human-in-the-loop learning and adaptation. Section 7 reviews simulation, sim-to-real transfer, digital twin validation, and simulation-assisted feedback. Section 8 synthesizes methodological trade-offs, failure modes, benchmark limitations, and future research directions. Section 9 presents the conclusions.

2. Review Methodology

This review adopts a structured scoping approach to identify, classify, and analyze studies relevant to language-guided dexterous grasping with human-in-the-loop adaptation. The reporting was informed by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses extension for Scoping Reviews [95], and the literature-search documentation was prepared with reference to relevant PRISMA-S items [96]. Study identification, screening, and final inclusion are summarized in a PRISMA-style flow diagram based on PRISMA 2020 [97].

The review methodology was designed to accommodate the interdisciplinary nature of the topic, which spans language-conditioned robotic manipulation, dexterous grasp generation, human-in-the-loop robot learning, tactile and multimodal perception, and simulation-based validation. Because relatively few studies address all of these components simultaneously, the classified corpus includes both studies that directly investigate language-guided dexterous grasping and studies that provide transferable methodological evidence for one or more constituent components.

After defining the scope and research questions, relevant literature was identified through database searches and citation tracing. The retrieved records were screened according to the eligibility criteria, and the included studies were subsequently analyzed and organized across the four overlapping methodological directions of the review.

2.1. Scope and Research Questions

The scope of this review is organized around four research directions: language-guided robotic manipulation, dexterous grasping, human-in-the-loop learning and adaptation, and simulation-based validation. These directions were selected because language-guided dexterous grasping cannot be reduced to a direct mapping from a natural-language instruction to a robot action. A complete system must interpret the instruction, identify task-relevant objects and object parts, generate physically admissible multi-fingered grasp candidates, account for semantic-alignment and functional-suitability requirements, incorporate user feedback when necessary, verify the resulting action before execution, and monitor it during physical execution.

The review primarily considers robotic grasping and manipulation methods in which natural language, vision–language representations, task descriptions, human feedback, or simulation-based evaluation influence grasp generation, grasp selection, or execution. Studies involving parallel-jaw grippers were also considered when they introduced language-grounding, vision–language–action, human-feedback, or validation mechanisms that are methodologically relevant to dexterous grasping. Their limitations with respect to high-dimensional multi-fingered control are explicitly considered in the subsequent analysis.

Studies of human hand–object interaction were selectively included when they provided representations, demonstrations, affordance annotations, contact information, or datasets that can support robotic dexterous grasping. Earlier foundational studies were also eligible when they provided substantive methodological evidence concerning grasp

stability, learning from demonstration, preference learning, or sim-to-real transfer and contributed directly to at least one research question. In contrast, secondary reviews, review-reporting guidelines, mathematical or statistical foundations, and software or platform documentation were retained as manuscript references where necessary but were not treated as classified-corpus studies.

Based on this scope, the review addresses the following research questions (RQs):

1. How are natural-language instructions grounded into target objects, object parts, affordances, prohibited contact regions, and task-dependent grasp constraints?
2. How are dexterous grasps represented, generated, and evaluated under kinematic, geometric, contact, semantic, and functional requirements?
3. How are demonstrations, corrections, preferences, evaluative feedback, and physical interventions incorporated into dexterous grasp generation and adaptation?
4. How are physics simulation, sim-to-real transfer, and digital twin environments used to verify grasp constraints, estimate execution outcomes, or improve language-conditioned dexterous grasping systems?
5. What methodological, data-related, evaluation, and deployment limitations currently prevent the development of scalable and reliable language-guided dexterous grasping systems?

These research questions define the analytical structure of the review. RQ1 is primarily addressed in Section 4 through the discussion of language reasoning, visual grounding, and vision–language–action policies. RQ2 is addressed in Sections 3 and 5 through the formulation and comparison of dexterous grasp representations and generation methods. RQ3 is examined in Section 6, while RQ4 is addressed in Section 7. The limitations identified in relation to RQ5 are synthesized in Section 8.

2.2. Search Strategy and Information Sources

To strengthen reproducibility and audit the coverage of the review corpus, supplementary structured searches were conducted in Scopus and the IEEE Xplore Digital Library on 14 July 2026. The searches were restricted to records published from 2021 to 2026, inclusive, with 2026 records limited to those indexed by the search date.

Four concept-specific searches were conducted separately in each database. Searches on language-guided robotic manipulation covered language conditioning, vision–language grounding, open-vocabulary perception, semantic grasping, affordance reasoning, and vision–language–action models. Searches on dexterous grasping covered multi-fingered grasp generation, functional grasping, contact-rich manipulation, and dexterous-hand control. Human-in-the-loop searches covered demonstration, language correction, preference learning, evaluative feedback, and physical intervention. Searches on simulation and validation covered physics simulation, sim-to-real transfer, real-to-sim-to-real learning, digital twins, execution verification and validation.

The four Scopus searches returned 2048 records, while the four IEEE Xplore searches returned 1445 records, resulting in 3493 records before deduplication. The complete database-specific search strings, query identifiers, search fields, applied filters, publication-year limits, search date, result counts, and archived export filenames are provided in Supplementary Table S1.

Google Scholar was used only during the preliminary exploration of terminology and potentially relevant literature. A candidate-level search log and a predefined result-depth stopping rule were not retained for this exploratory phase. Google Scholar was therefore not used to derive the quantitative identification, screening, or inclusion counts reported in this review; these counts are based on the archived Scopus and IEEE Xplore searches

and verified citation relationships. The overall identification, deduplication, screening, citation-searching, and inclusion process is summarized in Figure 1.

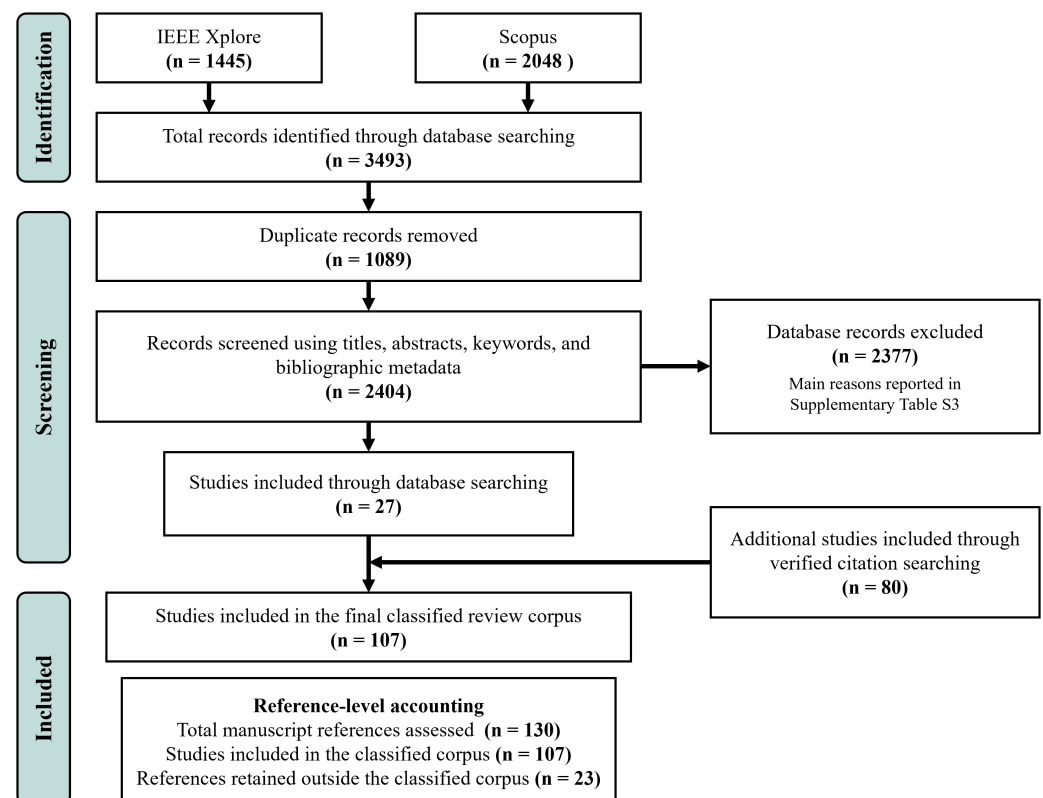


Figure 1. PRISMA-style flow diagram of study identification, screening, citation searching, and final corpus construction. The reference-level accounting distinguishes the 107 studies included in the classified review corpus from the 23 references retained outside the corpus among the 130 manuscript references.

2.3. Eligibility Criteria and Study Selection

The Scopus and IEEE Xplore search results were merged and deduplicated before screening. Duplicate records were identified primarily through exact DOI or arXiv-identifier matches. When persistent identifiers were unavailable, titles were compared after standardizing letter case, punctuation, hyphenation, and whitespace. Author names, publication year, venue, and publication status were additionally examined when conference papers, journal articles, or preprints appeared to represent different versions of the same study. After 1089 duplicate records were removed, 2404 unique records remained for screening.

The unique records were screened using titles, abstracts, author keywords, index keywords or IEEE terms, and bibliographic metadata. When the available metadata was insufficient to determine eligibility, the corresponding publication was examined for additional information. Each record was assigned an inclusion or exclusion decision according to the eligibility criteria. Within each query-defined area, studies were retained when they provided a distinct method, representation, feedback mechanism, validation strategy, dataset, or evaluation perspective relevant to at least one research question. Records that repeated an already represented methodological contribution without adding a distinct contribution to the review questions were excluded. Of the 2377 excluded records, 2166 did not provide a distinct methodological contribution beyond a representative study already retained, 81 were reviews, editorials, corrections, or other non-research documents, and 55 were outside the scope of robotic manipulation or grasping. The remaining exclusions comprised 31 records without a relevant human-feedback contribution, 21 human hand-object studies without transferable robotic relevance, 12 records without a simulation,

sim-to-real, digital twin, or validation contribution, eight records without a dexterous, multi-fingered, tactile, or otherwise transferable grasping contribution, and three secondary or contextual sources retained outside the classified corpus. The complete record-level decisions and exclusion reasons are provided in Supplementary Table S3.

A study was eligible for inclusion in the classified review corpus when it satisfied at least one of the following criteria:

- It proposed or evaluated a language-guided, language-conditioned, or vision-language-based robotic manipulation method;
- It addressed robotic dexterous grasp generation, multi-fingered grasp planning, contact-rich grasp execution, or functional grasping;
- It introduced a human-feedback mechanism relevant to robotic manipulation, including demonstration, correction, language feedback, preference, evaluative feedback, or physical intervention;
- It presented a simulation, sim-to-real, real-to-sim-to-real, or digital twin strategy applicable to robotic grasping, policy learning, or execution verification and validation;
- It introduced a dataset, benchmark, representation, or foundational formulation directly relevant to dexterous grasping, semantic or functional grasp selection, tactile perception, or human-guided adaptation.

Parallel-gripper studies were not excluded solely because they did not use dexterous robotic hands. Such studies were retained when their language-grounding, vision-language-action, feedback, action-generation, or validation mechanisms provided a distinct and transferable contribution to dexterous grasping. Studies of human hand-object interaction were similarly retained when their representations, demonstrations, contact annotations, or datasets contributed directly to one or more research questions.

Secondary reviews, review-reporting guidelines, mathematical or statistical foundations, and software or platform documentation were not treated as classified-corpus studies. These references were retained in the manuscript when required to position the review, support the analytical formulation, document the review methodology, or describe research infrastructure.

Recent preprints were selectively retained when they introduced directly relevant methods, datasets, platforms, or rapidly evolving technical developments and when no formally published version was available at the final search date. When both a preprint and a published version of the same study were available, the published version was used preferentially. Preprint-only studies appearing in comparative tables are marked with a superscript dagger.

Following database screening, 27 studies were included directly through the Scopus and IEEE Xplore searches. An additional 80 studies were included through verified backward or forward citation searching, resulting in a final classified review corpus of 107 studies.

Among the 130 references cited in the final manuscript, 107 studies constituted the classified review corpus. The remaining 23 references were retained outside the corpus because they served as secondary reviews, review-reporting guidelines, mathematical or statistical foundations, or software and platform documentation. These references were not included in the final study count, category counts, or direct-versus-supporting evidence totals. Supplementary Table S2 reports the study-level coding, evidence classification, and inclusion rationale for the 107 studies in the classified review corpus.

The eligibility criteria and coding framework were jointly discussed and established by both authors before screening. The first author conducted the database-record screening, made the initial inclusion and exclusion decisions, extracted the study-level information, and performed the initial category and evidence coding. The corresponding author reviewed the screening outcomes and verified the final study selection, category assignments,

and evidence classifications. Ambiguous cases were resolved through discussion between the authors.

2.4. Data Extraction and Classification

For each included study, information was extracted according to a common set of analytical dimensions. These dimensions were selected to support comparison across studies that use different robot embodiments, input modalities, learning paradigms, datasets, and evaluation settings.

The extracted information included:

- Methodological role and problem scope;
- Input modalities, including language, RGB images, depth images, point clouds, tactile sensing, and proprioception;
- Output representation, including task plans, end-effector actions, dexterous hand configurations, contact maps, grasp candidates, and policy trajectories;
- Grasp or action representation;
- Physical constraints and grasp-quality criteria;
- Semantic, affordance-based, or functional conditioning;
- Type of human feedback, when applicable;
- Learning, optimization, or generation paradigm;
- Simulation platform and sim-to-real strategy;
- Evaluation setting, including simulation-only and real-robot validation;
- Reported evaluation metrics;
- Principal limitation for language-guided dexterous grasping.

The evidence relationship of each included study was classified as direct or supporting. Direct evidence was defined as studies that used natural-language instructions, or multimodal inputs explicitly conditioned on language, to generate, select, or execute grasps with a dexterous or multi-fingered robotic hand. Supporting evidence was defined as studies that did not address all three elements—language conditioning, a dexterous or multi-fingered robotic embodiment, and grasp generation, selection, or execution—but provided a transferable methodological contribution to one or more constituent components of the focal problem.

The extracted information was used to organize the literature into four major and partially overlapping directions: language-guided robotic manipulation, dexterous grasping, human-in-the-loop learning, and simulation-based validation. Because the categories were not mutually exclusive, an individual study could be assigned to more than one direction. Figure 2 presents the resulting organizational taxonomy.

Within the first direction, language-guided robotic manipulation studies were classified into large-language-model-based task reasoning, vision–language grounding, and vision–language–action policies. Dexterous grasping studies were classified according to contact and configuration representations, optimization-based planning, probabilistic and generative modeling, functional and language-guided grasping, tactile and multimodal sensing, and dataset or evaluation design.

Human-in-the-loop studies were classified according to the form of feedback provided by the user, including demonstration, corrective action, language correction, preference, evaluative feedback, physical intervention, and simulation-preview feedback. Simulation-related studies were classified according to their use of physics simulation, manipulation benchmarks, domain or dynamics randomization, sim-to-real adaptation, real-to-sim-to-real learning, digital twin validation, and simulation-assisted human feedback.

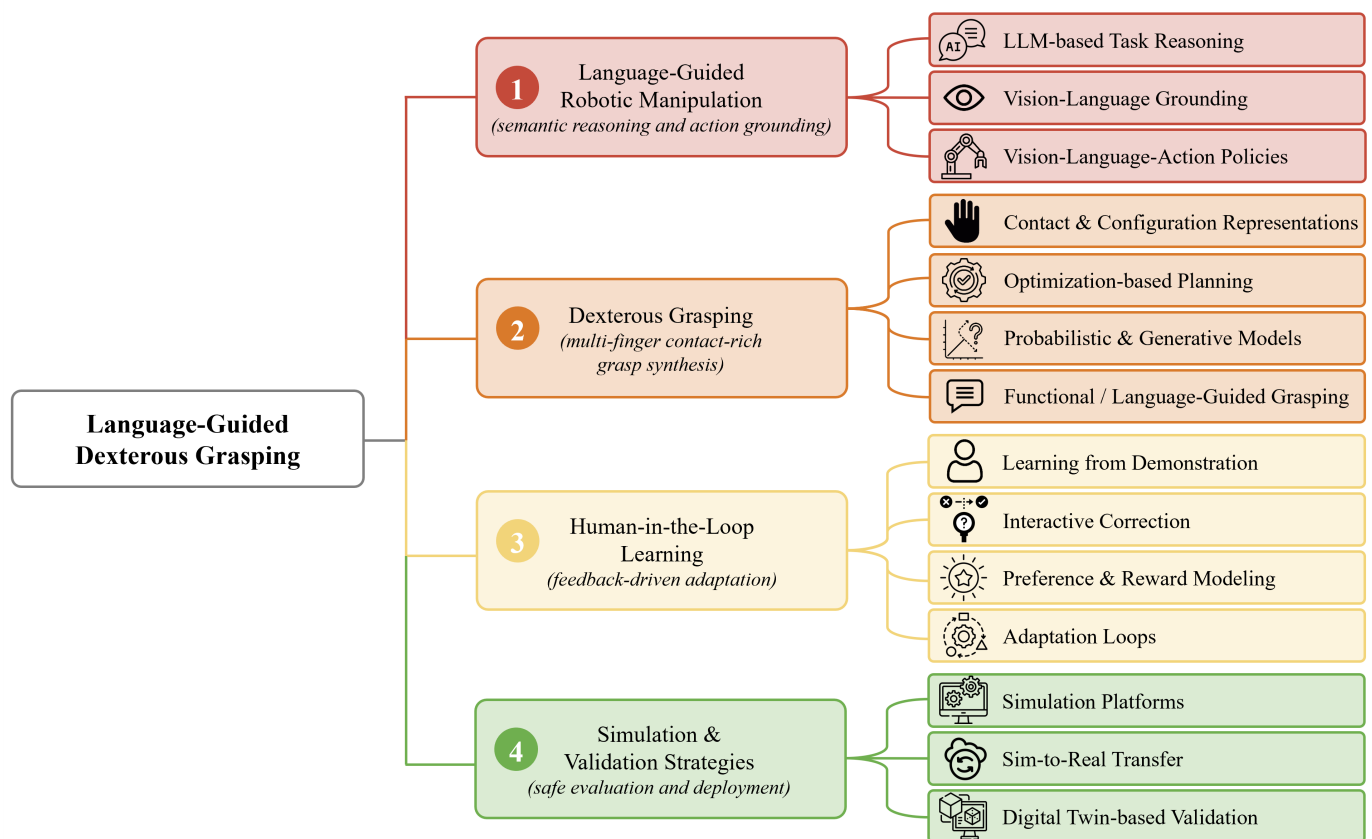


Figure 2. Organizational taxonomy used to classify the literature reviewed in this study. The taxonomy groups the literature into four overlapping methodological directions and their representative subtopics.

Among the 107 studies, 42 were classified under language-guided robotic manipulation, 45 under dexterous grasping and hand–object interaction, 18 under human-in-the-loop learning and feedback, and 42 under simulation, sim-to-real transfer, or digital twin validation. Because the categories overlap, the sum of these counts exceeds the size of the final corpus.

Five studies satisfied the direct-evidence definition, whereas 102 studies were classified as supporting evidence. The complete study-level evidence classifications, supporting-evidence subtypes, and inclusion rationales are provided in Supplementary Table S2. Supplementary Table S1 reports the complete database search strategies, Supplementary Table S3 reports the record-level screening decisions and exclusion reasons, and Supplementary Table S4 documents the citation-searching provenance of the citation-identified studies.

The resulting classification provides the basis for the organization of the remainder of the review. Section 3 introduces a common problem formulation for comparing the reviewed methods. Sections 4–7 examine the principal technical directions, while Section 8 synthesizes cross-cutting methodological differences, failure modes, evaluation limitations, and future research directions.

3. Problem Formulation

This section introduces a common analytical formulation for comparing the methodological paradigms reviewed in this study. The formulation does not prescribe a single algorithm or fixed system architecture. Instead, it separates five decisions that are frequently conflated in language-guided dexterous grasping: candidate generation, admissibility

verification, evaluation of semantic alignment and functional suitability, multi-objective selection under uncertainty, and adaptation through human feedback.

Classical robotic grasping established analytical foundations for contact stability, closure properties, grasp synthesis, and grasp quality [1–3]. Physics-based tools subsequently enabled grasp candidates to be evaluated under different hand, object, and contact configurations [4]. More recent methods generate high-dimensional dexterous grasps from object geometry or learned distributions [18,19,24–26,28], while language-guided approaches additionally condition grasp generation on object parts, affordances, task requirements, or natural-language instructions [50,52–55]. Human-in-the-loop approaches further allow demonstrations, corrections, preferences, and interventions to modify the grasp-generation or selection process [60,62,64,66,68,69].

Throughout this review, the following terms are used with fixed meanings. Admissibility refers to membership in the admissible grasp set G_{adm} defined by the mandatory kinematic, reachability, collision, contact, and safety constraints in Equation (3). Semantic alignment refers to the degree to which the selected object, object part, contact region, or grasp type corresponds to the natural-language instruction, as represented by S_{sem} in Equation (7). Functional suitability refers to the degree to which the grasp supports the intended downstream task, as represented by S_{func} . Execution risk refers to a scalar measure of failure-related loss under a distribution of execution uncertainty, represented by $R_{\text{exec}}(g; P_{\xi})$. Verification refers to checking whether a candidate grasp, selected action, or learned policy satisfies explicitly specified requirements, including kinematic limits, reachability, collision, contact, force, and safety constraints. Verification may be performed analytically, numerically, or through simulation-based constraint checking. Validation refers to evaluating whether the verified system performs its intended task under representative operating conditions. Simulation and digital twins can provide validation evidence for the modeled environment, whereas real-robot trials provide evidence concerning physical deployment. Simulated success alone does not validate real-world performance. Neither verification nor validation is used here as a synonym for semantic alignment or functional suitability. These terms are used consistently in the subsequent sections, tables, and discussion to avoid conflating physical admissibility, semantic interpretation, downstream task usefulness, user-dependent preference, and uncertainty-aware execution outcomes.

3.1. Language-Conditioned Dexterous Grasping

Let O denote object or scene representations, x a multimodal observation, I a natural-language instruction, T the intended task or downstream use, and h human feedback when available. The observation x may contain RGB images, depth images, point clouds, robot proprioception, and tactile measurements. These modalities provide complementary information about object geometry, hand configuration, contact formation, pressure, slip, and post-contact stability [31,98].

A common probabilistic abstraction of language-conditioned dexterous grasp generation is

$$p_{\theta}(g \mid O, x, I, T, h), \quad (1)$$

where g denotes a dexterous grasp candidate and θ denotes the parameters of a learned model, optimizer, policy, or generative process. Equation (1) encompasses deterministic predictors as a limiting case, as well as probabilistic proposal networks, diffusion-based generators [26], vision–language–grasp models [50], and hierarchical methods [51,53] that first infer semantic grasp intent and subsequently generate a compatible hand configuration.

The task variable T is distinguished from the literal instruction I . For example, the instruction “grasp the scissors” does not fully specify whether the object will subsequently be used for cutting, placed into storage, or handed to another person. These tasks impose

different requirements on which regions may be contacted and which regions must remain accessible. Functional grasping studies similarly demonstrate that physical stability alone does not determine whether a grasp is suitable for the intended manipulation task [55–58].

The feedback variable h represents information that supplements or corrects the initial instruction. It may include a language correction, demonstrated pose, approval, rejection, pairwise preference, or physical intervention. For example, “do not touch the blade” may introduce a mandatory exclusion constraint, whereas “grasp closer to the handle” may alter the relative ranking among otherwise admissible grasps [64–66].

Equation (1) is used in the remainder of this review as a common reference for comparing what information each methodology family conditions on. Section 4 examines how I , O , and x are processed by language, vision–language, and vision–language–action models. Section 5 compares the representations used for g , while Section 6 examines how h modifies generation, ranking, or policy adaptation.

3.2. Grasp Representation and Physical Admissibility

A dexterous grasp requires a richer representation than the end-effector pose and opening width typically used for parallel-jaw grasping. For analytic comparison, a grasp candidate is represented as

$$g = (T_h, q, c, f), \quad (2)$$

where $T_h \in SE(3)$ denotes the global hand or palm pose, $q \in \mathcal{Q}$ denotes the finger-joint configuration, $c \in \mathcal{C}$ denotes contact points, contact regions, or contact maps, and $f \in \mathcal{F}$ denotes optional force-, wrench-, pressure-, or compliance-related variables.

Equation (2) provides a common superset for comparing grasp representations. Individual methods may predict only a subset of its components. Some methods directly generate T_h and q , whereas others generate contact maps c and subsequently recover a compatible hand configuration through optimization or refinement. Force-related variables f may be explicitly predicted, measured through tactile sensing, or evaluated indirectly through analytic metrics or simulation [18,19,24–26,31].

A generated candidate is not necessarily executable. Physical, task-dependent, and safety-critical requirements are therefore combined into an admissible grasp set:

$$\mathcal{G}_{\text{adm}} = \left\{ g \mid \begin{array}{l} C_{\text{kin}}(g) \leq 0, \\ C_{\text{reach}}(g; O) \leq 0, \\ C_{\text{col}}(g; O) \leq 0, \\ C_{\text{contact}}(g; O) \leq 0, \\ C_{\text{safe}}(g; I, T, h) \leq 0 \end{array} \right\}. \quad (3)$$

Each constraint function in Equation (3) is defined such that a non-positive value indicates satisfaction of the corresponding requirement, whereas a positive value indicates a constraint violation. The constraint values do not necessarily represent physical quantities that must themselves be negative. Instead, each function measures the difference between the required threshold and the value achieved by a candidate grasp. This non-positive convention is a standard form used in constrained optimization.

For example, the joint configuration q must satisfy

$$q_{\min} \leq q \leq q_{\max}. \quad (4)$$

This requirement can be expressed through the two inequality constraints

$$q_{\min} - q \leq 0, \quad q - q_{\max} \leq 0. \quad (5)$$

Both expressions are non-positive when the joint configuration remains within its allowable range. If a joint exceeds either limit, the corresponding constraint becomes positive and indicates the magnitude of the violation.

Similarly, if the minimum distance $d(g)$ between the robot and an unintended obstacle must be greater than or equal to a required clearance $d_{\min}$, a collision-related constraint can be defined as

$$d_{\min} - d(g). \quad (6)$$

The condition in Equation (3) $C_{\text{col}}(g; O) \leq 0$ is equivalent to $d(g) \geq d_{\min}$. A negative value indicates that the required clearance condition is satisfied, a value of zero indicates that the candidate lies on the constraint boundary, and a positive value indicates insufficient clearance or a risk of collision.

A minimum grasp-quality requirement can be represented in the same way. If $Q(g)$ denotes a physical grasp-quality measure and $Q_{\min}$ denotes the minimum acceptable quality, then $Q_{\min} - Q(g) \leq 0$ requires $Q(g) \geq Q_{\min}$. Depending on the study, $Q(g)$ may represent force closure, grasp-wrench-space quality, contact stability, slip resistance, lift success probability, or a learned grasp-quality score [1–4].

Within Equation (3), C_{kin} includes joint limits and hand-kinematic constraints; C_{reach} represents arm- or wrist-level reachability; C_{col} represents self-collision, environmental collision, and unintended object penetration; C_{contact} represents contact, friction, closure, or minimum stability conditions; and C_{safe} represents mandatory task-, language-, or user-dependent restrictions.

The constraint functions provide common notation for comparing how different methods represent admissibility requirements. Classical grasp-analysis methods explicitly formulate contact and closure constraints [1–3], whereas simulation- and learning-based approaches may approximate the same requirements through collision checking, physical roll-out, learned success classifiers, contact-quality models, or iterative refinement [4,18,24,27].

The distinction between a generated grasp and an admissible grasp is central to this review. A learned model may assign high probability to a candidate that resembles its training data but violates the joint limits of another robot hand, penetrates the object surface, produces unstable contacts under the current friction conditions, or conflicts with a language-dependent safety requirement. Candidate generation and admissibility verification should therefore be treated as related but distinct stages.

The set $\mathcal{G}_{\text{adm}}$ is generally non-convex because collision, contact, friction, and kinematic constraints are nonlinear. Changes in finger contact modes may introduce discontinuities, and different contact combinations may form disconnected feasible regions. Consequently, gradient-based optimization may be sensitive to initialization and converge to different local solutions. Conversely, learned or generative models may assign probability mass to physically invalid regions or fail to represent low-frequency but task-relevant grasp modes.

If $\mathcal{G}_{\text{adm}}$ is non-empty and compact and a scalar objective is continuous over this set, at least one global optimum exists. These conditions do not guarantee that the optimum is unique or that it can be computed efficiently [99]. In practical dexterous grasping, global optimization remains difficult because the admissible set is high-dimensional, non-convex, and dependent on uncertain contact and object parameters.

An empty admissible set also has an important operational interpretation. It may indicate a conflict between the instruction and the physical capabilities of the robot rather than a failure of grasp generation alone. For example, a language instruction may require contact with a small target region that the current hand cannot reach without violating a joint limit or colliding with the object. In such cases, the system should request clarification, change the approach direction, relax a non-critical preference, use a different end effector,

or reject the action. A mandatory physical or safety violation should not be compensated for by increasing a soft semantic-alignment or functional-suitability score.

Equation (3) is used later to distinguish grasp generation from physical and safety admissibility verification. Section 5 compares how dexterous grasping methods represent and approximate the admissible region, while Section 7 examines which constraints can be evaluated using simulation or digital twins and which require calibration through real-world execution.

3.3. Physical, Semantic, Functional, and User-Dependent Objectives

Physical admissibility should be distinguished from grasp quality. Equation (3) determines whether a grasp candidate satisfies mandatory kinematic, reachability, collision, contact, and safety requirements. However, multiple candidates may remain admissible while differing substantially in physical stability, semantic correspondence, downstream functionality, user preference, and execution risk.

Let ξ denote a realization of execution uncertainty and P_ξ denote its probability distribution. For analytical comparison, this review decomposes grasp evaluation into the following objective vector:

$$\left[S_{\text{phys}}(g; O, x), S_{\text{sem}}(g; I, O, x), S_{\text{func}}(g; I, T, O), S_{\text{user}}(g; h), -R_{\text{exec}}(g; P_\xi) \right]. \quad (7)$$

Equation (7) is a review-level analytical synthesis rather than an objective function taken from a single prior study. Its purpose is to distinguish evaluation criteria that are frequently combined into one composite score without sufficient explanation. Each component is interpreted as follows.

3.3.1. Physical Quality

$S_{\text{phys}}(g; O, x)$ evaluates the physical quality of a candidate that has already satisfied the minimum admissibility conditions. It may represent force closure, grasp-wrench-space quality, contact distribution, slip margin, disturbance resistance, lift success, or a learned probability of stable execution [1–4,18,27].

The distinction between physical admissibility and physical quality is important. A candidate may satisfy a minimum stability threshold and therefore belong to $\mathcal{G}_{\text{adm}}$, while another admissible candidate may provide a larger stability margin or greater resistance to perturbation. Thus, $C_{\text{contact}}(g; O) \leq 0$ in Equation (3) represents a minimum acceptance condition, whereas S_{phys} ranks candidates that have already passed this condition.

3.3.2. Semantic Alignment

S_{sem} measures whether the selected object, object part, contact region, or grasp type is consistent with the natural-language instruction. It may be operationalized through object- or part-grounding accuracy, overlap between predicted contact regions and language-referred regions, instruction-grasp compatibility, or similarity scores obtained from vision-language representations [50,52,100–102].

Semantic alignment does not guarantee physical admissibility. A model may correctly identify the handle specified by an instruction but generate a hand configuration that violates joint limits, penetrates the object, or produces unstable contacts. Conversely, a physically stable grasp may contact an object region that contradicts the instruction.

3.3.3. Functional Suitability

S_{func} evaluates whether a grasp supports the intended downstream task T . A functionally appropriate grasp should preserve access to task-critical object regions and en-

able subsequent actions such as cutting, pouring, handover, assembly, insertion, or tool use [55–58].

Semantic alignment and functional suitability are related but not identical. Semantic alignment concerns whether the grasp corresponds to the expressed instruction, whereas functional suitability concerns the consequences of the grasp during subsequent task execution. For example, a grasp may correctly follow an instruction to hold a screwdriver by its handle but still be functionally unsuitable if the wrist orientation prevents the tool from being used in the required direction.

3.3.4. User-Dependent Preference

S_{user} represents preferences that cannot be fully determined from physical, semantic, or functional criteria. These preferences may concern comfort, predictability, handedness, personal safety margins, preferred handover orientation, or the relative importance assigned to multiple valid grasp alternatives.

User-dependent utility may be inferred from demonstrations, approvals, rejections, pairwise comparisons, language corrections, or previous interaction histories [61–63,103]. Unlike physical quality, this component may vary across users even when the object and task remain unchanged.

3.3.5. Execution Risk

Relevant uncertainty sources include errors in object pose, incomplete geometry, uncertain friction and mass distribution, actuator error, perception noise, language-grounding error, and simulation-to-reality discrepancy. The term $R_{\text{exec}}(g; P_{\xi})$ in Equation (7) denotes a scalar risk measure defined over the execution-loss distribution induced by P_{ξ} . Depending on the application, it may be instantiated as a failure probability, an expected execution loss, or an adverse-tail risk measure.

Execution risk may account for collision, slip, excessive contact force, trajectory failure, object damage, or failure to maintain the grasp during downstream manipulation. Simulation, domain randomization, and digital twin methods can support its estimation [81,82,84,85]. However, these methods do not independently determine whether a grasp is semantically aligned, functionally suitable, or preferred by a particular user.

The negative sign in Equation (7) indicates that execution risk is minimized, whereas the other objective components are maximized.

3.3.6. Hard Constraints and Soft Objectives

A requirement may be represented either as a hard constraint in Equation (3) or as a soft objective in Equation (7). The appropriate choice depends on the task and the consequence of violation.

For example, the instruction “do not touch the blade” should generally be represented through C_{safe} , because a prohibited contact should not be compensated for by strong performance in another objective. In contrast, a preference such as “prefer a grasp near the center of the handle” may be represented through S_{sem} , S_{func} , or S_{user} , because alternative contact locations may remain acceptable.

The same criterion may change from a soft objective to a hard constraint depending on the context. Avoiding the opening of an empty cup may be a preference, whereas avoiding the opening of a cup containing liquid may be mandatory. Consequently, language interpretation must identify not only the relevant object region but also whether the expressed requirement represents a prohibition, a minimum requirement, or a relative preference.

The components of Equation (7) may have different numerical ranges and statistical interpretations. Therefore, weighted aggregation requires an explicit normalization, calibration, or utility-mapping procedure. Because the selected transformation can affect the

final ranking, studies using composite grasp scores should report the original component values, their calibration procedure, and the rule used to combine them.

Table 2 summarizes representative measures and validation procedures for the five components of Equation (7). The listed measures are illustrative rather than universal because their applicability depends on the robot embodiment, object set, task definition, and evaluation environment.

Table 2. Representative operational measures and validation methods for the objective components used in language-guided dexterous grasp evaluation.

Component	Operational Measures	Verification or Validation Evidence	Main Limitation
Physical quality	Force closure, wrench-space margin, slip margin, lift success, and perturbation resistance	Analytic evaluation, physics simulation, repeated lift tests, or real-robot perturbation trials	Sensitive to contact models, friction estimates, object geometry, and robot-hand embodiment
Semantic alignment	Object- or part-grounding accuracy, contact-region overlap, and instruction-grasp compatibility	Annotated object-part datasets, language-grounding benchmarks, or human evaluation	Correct semantic grounding does not guarantee a physically executable grasp
Functional suitability	Downstream task success, accessibility of task-critical regions, and preservation of object affordances	Task execution involving tool use, pouring, handover, insertion, or assembly	Requires task-specific evaluation and cannot generally be inferred from static grasp quality
User preference	Approval rate, pairwise choice, correction frequency, and inferred user-conditioned utility	Controlled user studies, repeated interaction sessions, or comparison with user demonstrations	May vary across users, tasks, and sessions and may be affected by fatigue or inconsistency
Execution risk	Collision, slip, excessive-force events, failure under parameter variation, and tail execution loss	Repeated simulation trials followed by calibration against real-robot outcomes	Simulation-derived risk may be inaccurate when real-world failure modes are not reproduced

3.4. Multi-Objective Grasp Selection

The components introduced in Equation (7) may conflict with one another. For example, a grasp that provides high physical stability may obstruct a task-critical object region, whereas a functionally suitable grasp may use a smaller contact area and become more sensitive to perturbations. A grasp preferred by a user may also differ from the candidate that maximizes an analytical grasp-quality metric. Consequently, reducing all evaluation criteria to a single unexplained score can conceal meaningful trade-offs.

Multi-objective optimization provides a framework for analyzing these trade-offs without assuming that one fixed scalar objective is appropriate for every object, task, robot embodiment, and user [99].

3.4.1. Pareto-Based Selection

Let g_a and g_b denote two candidates in the admissible grasp set $\mathcal{G}_{\text{adm}}$. Candidate g_a Pareto-dominates candidate g_b when g_a performs at least as well as g_b in every objective and performs strictly better in at least one objective:

$$g_a \succ_P g_b \iff [S_i(g_a) \geq S_i(g_b) \text{ for all } i] \wedge [S_j(g_a) > S_j(g_b) \text{ for at least one } j]. \quad (8)$$

The corresponding Pareto-optimal grasp set is:

$$\mathcal{G}_P = \{g \in \mathcal{G}_{\text{adm}} \mid \nexists g' \in \mathcal{G}_{\text{adm}} \text{ such that } g' \succ_P g\}. \quad (9)$$

A candidate outside $\mathcal{G}_p$ can be removed because another admissible candidate performs at least as well in every objective and strictly better in at least one objective. The remaining candidates therefore represent meaningful trade-offs among the objective components in Equation (7).

Pareto analysis is useful when the relative importance of the objectives cannot be justified in advance. However, it generally produces multiple non-dominated candidates rather than one final grasp. An additional decision rule is therefore required to select among the remaining candidates.

3.4.2. Weighted Selection

A weighted sum provides one possible rule for selecting a single grasp candidate:

$$g^*(\mathbf{w}) = \arg \max_{g \in \mathcal{G}_{\text{adm}}} \sum_{i=1}^m w_i S_i(g), \quad (10)$$

Subject

$$w_i > 0, \quad \sum_{i=1}^m w_i = 1. \quad (11)$$

Here, m denotes the number of evaluation objectives, $S_i(g)$ denotes the value of objective i for candidate g , and w_i represents its relative importance.

When all weights are strictly positive, a global maximizer of the weighted objective is Pareto optimal. However, the converse does not necessarily hold. In particular, linear weighted selection may fail to recover Pareto optimal solutions located in non-convex regions of the objective space [99,104]. This limitation is relevant to dexterous grasping because nonlinear hand kinematics, discrete contact modes, collision constraints, and self-collision constraints commonly produce non-convex solution regions.

The final selection is also sensitive to the numerical scale of each objective. If one objective has a substantially larger numerical range than the others, it may dominate the combined score even when its assigned weight is relatively small. The interpretation of w_i therefore depends on how the corresponding objective value $S_i(g)$ is calibrated [105].

A fixed weight configuration may also fail to generalize across different objects, tasks, robot hands, users, and safety conditions. Weights should therefore be derived from an explicit source, such as task requirements, safety priorities, expert knowledge, learned user preferences, or context-dependent utility estimation.

Studies using weighted selection should report the definition and scale of each objective, the selected weights, the source of those weights, and the sensitivity of the final candidate ranking to alternative weight values.

Mandatory safety requirements should remain in the admissible grasp set rather than being represented only as penalties in the weighted objective. Otherwise, a serious collision, prohibited contact, or excessive-force condition could theoretically be offset by high physical, semantic, functional, or user-preference scores.

3.4.3. Risk-Sensitive Selection

Average success may be insufficient when grasp execution includes rare but severe failures. A candidate may succeed under most conditions but fail under a small subset of object-pose, friction, mass, perception, or actuator conditions.

A risk-sensitive selection rule can be represented as

$$g_{\text{risk}}^* = \arg \max_{g \in \mathcal{G}_{\text{adm}}} \left[\mathbb{E}_{\xi \sim P_{\xi}} [U(g, \xi)] - \beta \text{CVaR}_{\alpha}(L(g, \xi)) \right]. \quad (12)$$

In this formulation, $\xi \sim P_\xi$ represents possible execution conditions drawn from the uncertainty distribution P_ξ . The function $U(g, \xi)$ denotes the utility of grasp g under a particular condition, whereas $L(g, \xi)$ denotes the corresponding execution loss. The conditional Value-at-Risk (CVaR) term measures the average loss among adverse execution outcomes, and $\beta \geq 0$ controls the balance between utility and risk [106]. Equation (12) converts the multiple objectives in Equation (7) into a scalar selection criterion. In this formulation, CVaR is used as one possible measure of the execution-risk component $R_{\text{exec}}(g; P_\xi)$ rather than as an additional independent objective.

When $\beta = 0$, selection depends only on expected utility. Increasing β gives greater importance to low-probability but high-cost failures. However, an excessively large value of β may produce overly conservative grasp selection and reject candidates that otherwise provide high expected performance. This criterion distinguishes average success from robustness to rare but severe failures.

Physics simulation and digital twin environments can support risk estimation by repeatedly evaluating candidates under variations in object pose, friction, mass, actuator behavior, and perception noise. However, simulation-derived risk estimates are meaningful only when the simulated failure modes and outcome distributions correspond sufficiently closely to those observed during real-robot execution [81,82,84,85].

Figure 3 summarizes the relationships among candidate generation, admissibility verification, multi-objective evaluation, and final grasp selection introduced in this section. Multimodal observations, language instructions, object or scene information, downstream-task context, and human feedback condition the generation of multiple grasp candidates. The generated candidates are restricted to $\mathcal{G}_{\text{adm}}$ according to mandatory physical and safety requirements and then evaluated using the objective components in Equation (7) before a final grasp is selected.

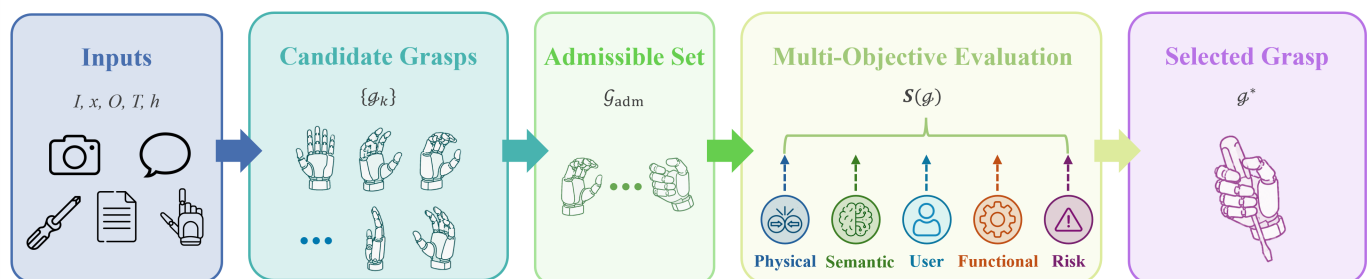


Figure 3. The language instruction I , multimodal observation x , object or scene representation O , downstream-task context T , and human feedback h condition the generation of candidate grasps $\{g_k\}$. Candidates satisfying the mandatory kinematic, reachability, collision, contact, and safety constraints constitute the admissible grasp set $\mathcal{G}_{\text{adm}}$. The admissible candidates are then compared using the objective components in Equation (7), and a final grasp g^* is selected. The figure summarizes the analytical framework used to compare the reviewed methods. Different colors distinguish the major stages of the framework, arrows indicate the processing flow, and ellipses represent omitted intermediate grasp candidates.

These formulations address different parts of the grasp-selection problem. Constrained optimization emphasizes admissibility and safety, Pareto analysis preserves trade-offs, weighted selection produces a final ranking, generative methods provide candidate diversity, risk-sensitive methods account for uncertain failures, reinforcement learning supports sequential control, and preference-based methods incorporate user-dependent utility. No single formulation is universally superior, and the appropriate method depends on the task requirements and deployment conditions. Table 3 compares the principal grasp-selection formulations according to their decision principles, strengths, and limitations.

Table 3. Comparison of grasp-selection formulations.

Formulation	Decision Principle	Principal Strength	Principal Limitation
Constrained optimization	Optimizes only within an explicitly defined admissible grasp set	Provides interpretable physical and safety requirements	May be affected by non-convexity, discontinuities, and local optima
Weighted selection	Combines objective values using task- or user-dependent weights	Provides a simple rule for selecting one candidate	Sensitive to objective scale and weight selection
Pareto optimization	Retains candidates that are not dominated across the objectives	Preserves meaningful trade-offs without fixed weights	Requires an additional rule for final selection
Probabilistic or generative modeling	Generates multiple candidate grasps from a learned or probabilistic distribution	Can generate diverse grasp configurations	Generated candidates may violate physical or safety constraints
Risk-sensitive optimization	Balances expected utility against uncertain high-loss outcomes	Distinguishes average performance from tail risk	Requires a calibrated uncertainty and failure model
Preference-based selection	Infers user-dependent utility from comparisons or feedback	Supports task- and user-specific personalization	Affected by sparse, inconsistent, or changing feedback

3.5. Human Feedback and Adaptation

Human feedback can affect different components of a grasping system and should not be treated as one undifferentiated supervision signal. Demonstrations, language corrections, pairwise preferences, approvals, rejections, and physical interventions provide different types of information and should influence the system in different ways.

3.5.1. Levels of Human Feedback

Human feedback can be classified according to the component of the grasping system that it modifies.

- Constraint-level feedback modifies mandatory requirements. For example, the instruction “do not grasp the table” should remove every candidate that contacts the prohibited region.
- Target-level feedback modifies the desired palm pose, finger configuration, contact region, wrist orientation, or trajectory. For example, “move the grasp closer to the center of the handle” changes the desired contact location without necessarily invalidating every alternative grasp.
- Ranking-level feedback changes the ordering of otherwise admissible candidates. A user may prefer one grasp because it is more predictable, safer during handover, more comfortable, or more appropriate for a subsequent task.
- Policy-level feedback updates the grasp-generation or execution model. Demonstrations and repeated corrections may therefore influence future behavior rather than only the current grasp decision.

These feedback levels are not equivalent. Constraint-level feedback can immediately remove an unsafe candidate, whereas policy-level adaptation may require repeated examples before its effect generalizes to unseen objects or states.

A mandatory correction received at interaction step t can be represented as an update of the admissible grasp set:

$$\mathcal{G}_{\text{adm}}^{t+1} = \mathcal{G}_{\text{adm}}^t \cap \{g \mid C_h(g; h_t) \leq 0\}. \quad (13)$$

Here, h_t denotes the feedback received at interaction step t , and $C_h(g; h_t)$ represents the additional constraint induced by that feedback. The condition

$$C_h(g; h_t) \leq 0 \quad (14)$$

indicates that candidate g satisfies the correction, whereas

$$C_h(g; h_t) > 0 \quad (15)$$

indicates that the candidate violates the correction.

For the correction “do not grasp the blade”, the updated admissible set excludes candidates that contact the prohibited region. By contrast, a softer correction such as “grasp closer to the center of the handle” may change candidate ranking without excluding every alternative grasp. A mandatory safety correction should therefore modify the admissible set, whereas a preference should generally modify the semantic, functional, or user-dependent evaluation of otherwise valid candidates.

3.5.2. Demonstration and Physical Interventions

Demonstrations provide positive examples of desired palm poses, finger configurations, contact patterns, trajectories, or action sequences [60,107–110]. Their effect can be interpreted as increasing the probability of demonstrated behavior within the grasp-generation model.

Physical interventions provide additional information because they occur when autonomous behavior becomes unsafe, unsuccessful, or undesirable. An intervention therefore provides two types of information: a corrective action indicating what the robot should do and an implicit negative signal indicating that the original action was unacceptable [111,112].

Intervention data are not generally sampled uniformly from the operating environment. Users are more likely to intervene in difficult, unfamiliar, or dangerous states. Consequently, intervention datasets may overrepresent failure-prone situations relative to the normal state distribution of the autonomous policy.

Demonstrations and interventions also differ in their informational content. A demonstration primarily indicates desired behavior, whereas an intervention additionally identifies a state–action pair that should be avoided. Treating these two forms of feedback as identical training samples may discard useful information about unsafe or undesirable behavior.

3.5.3. Pairwise Preference Modeling

When several admissible grasp candidates remain and no candidate dominates all alternatives, a user may provide a preference between two candidates.

The probability that candidate g_i is preferred to candidate g_j can be represented using a Bradley–Terry model [113]:

$$P_\phi(g_i \succ g_j \mid z) = \frac{\exp(R_\phi(g_i, z))}{\exp(R_\phi(g_i, z)) + \exp(R_\phi(g_j, z))}. \quad (16)$$

Here, z denotes the relevant context, which may include the object, observation, language instruction, downstream task, and user identity. The function $R_\phi(g, z)$ represents the learned user-dependent utility of grasp g under context z .

The model estimates relative rather than absolute utility. Adding the same constant c to both utility values does not change the preference probability:

$$\frac{\exp(R_\phi(g_i, z) + c)}{\exp(R_\phi(g_i, z) + c) + \exp(R_\phi(g_j, z) + c)} = \frac{\exp(R_\phi(g_i, z))}{\exp(R_\phi(g_i, z)) + \exp(R_\phi(g_j, z))}. \quad (17)$$

Therefore, pairwise comparisons determine relative preference but do not uniquely determine the absolute offset of the learned utility function.

This property becomes important when preference utility is combined with physical quality, semantic alignment, functional suitability, and execution risk. A reference value, calibration rule, or regularization procedure is required before the preference score can be meaningfully combined with the other objective components.

Preference modeling also commonly assumes that user preferences remain stable. In practice, preferences may change with the object, task, fatigue level, safety concern, handedness, physical capability, or interaction experience. Different users may also prefer different contact regions, wrist orientations, or handover configurations.

Sparse comparisons produce uncertain preference estimates, whereas contradictory comparisons may arise when differences between candidates are ambiguous or when user judgments change over time. Preference-based systems should therefore report the number of comparisons, agreement or consistency rates, uncertainty in the learned preference model, and performance on users or tasks not included during training.

3.5.4. Feedback Burden and Selective Querying

Human feedback imposes non-equivalent costs in supervision time, expertise, cognitive effort, task interruption, and latency. Section 6 formalizes these costs and reviews strategies for allocating feedback efficiently.

Table 4 compares the principal feedback types according to the level of the grasping system that they modify, the information they provide, and their main limitations.

Table 4. Comparison of human-feedback types and their roles in the common formulation.

Feedback Type	Feedback Level and Main Effect	Information Provided	Principal Limitation
Demonstration	Target- or policy-level feedback that modifies the desired grasp or future policy distribution	Example of a desired palm pose, finger configuration, contact pattern, trajectory, or action sequence	Requires substantial collection effort and may not sufficiently cover failure states
Corrective action	Target- or policy-level feedback that modifies an observed grasp, motion, or control action	Replacement pose, trajectory segment, or low-level corrective action	Requires the user to understand the robot state and may be difficult for high-dimensional finger control
Language correction	Constraint-, target-, or ranking-level feedback that modifies requirements or candidate ordering	Semantic or functional correction such as “avoid the blade” or “move toward the handle center”	May be ambiguous or difficult to convert into geometric, contact, or force requirements
Pairwise preference	Ranking-level feedback that estimates relative user-dependent utility through Equation (16)	Preferred candidate from a pair of grasps that already satisfy the mandatory constraints	Requires repeated queries and may produce sparse, inconsistent, or context-dependent judgments
Evaluative feedback or approval/rejection	Ranking- or policy-level feedback that indicates whether one proposed behavior is acceptable	Binary approval, rejection, scalar evaluation, or qualitative assessment	Provides limited information about which component of the rejected behavior should be changed
Physical intervention	Policy- and safety-level feedback that replaces unsafe behavior and identifies an undesirable state–action region	Corrective action together with implicit negative evidence for the original autonomous action	Requires continuous attention, introduces takeover delay, and is concentrated in difficult states
Simulation-preview feedback	Constraint-, target-, or ranking-level feedback applied before real execution	Approval, rejection, comparison, or revision based on a simulated candidate and predicted outcome	Depends on simulation fidelity and may increase interaction latency or induce user overtrust

The comparison in Table 4 is intended to avoid treating all human feedback as a single supervision signal. For the remainder of the review, the main distinction is whether feedback modifies the admissible grasp set, the target grasp or trajectory, the ranking among admissible candidates, or the future policy. This distinction is particularly important for dexterous grasping because users may need to judge finger configuration, contact location, wrist orientation, and downstream-task consequences simultaneously.

Accordingly, feedback efficiency should be evaluated together with grasp performance. Selective querying can reduce burden by requesting feedback only under ambiguity, high model uncertainty, close candidate trade-offs, or elevated execution risk. However, overly

restrictive querying may also prevent the system from receiving feedback in unfamiliar or safety-critical states. The later discussion therefore focuses on how feedback is routed and reused rather than repeating the feedback-type descriptions already summarized in Table 4.

4. Language-Guided Robotic Manipulation

Language-guided robotic manipulation includes several decision levels that affect different components of the common formulation introduced in Section 3. Large language models typically infer task intent, subgoals, constraints, or available skills; vision–language models associate linguistic concepts with objects, parts, affordances, and spatial regions; and vision–language–action models generate executable actions conditioned on multimodal observations and language. These outputs are not interchangeable. A symbolic task plan, an object-part mask, a six-degree-of-freedom end-effector command, and a dexterous grasp candidate g provide different information and satisfy different physical assumptions.

Most general LLM-, VLM-, and VLA-based studies reviewed in this section are treated as supporting evidence because they primarily use arm-level manipulation or relatively simple end effectors and do not directly generate, select, or execute multi-fingered robotic grasps. Their relevance lies in transferable language-grounding, spatial-reasoning, and action-generation mechanisms.

Equation (1) provides the common abstraction used throughout this section. Language-guided methods differ in how they process the object or scene representation O , multimodal observation x , instruction I , downstream task T , and human feedback h . Their outputs may alter the conditional distribution over grasp candidates, provide information for S_{sem} and S_{func} , or introduce restrictions that should be transferred to C_{safe} . Semantic competence and action-generation capability, however, do not automatically establish that a generated candidate belongs to $\mathcal{G}_{\text{adm}}$.

A useful distinction is whether a method maps the conditions in Equation (1) directly to a grasp or introduces an intermediate semantic, spatial, or skill-level representation. Let ζ denote such an intermediate representation. Depending on the method, ζ may represent an object part, affordance region, grasp type, robot skill, task subgoal, three-dimensional value map, or high-level action. A hierarchical conditional model can be represented as

$$p_{\theta}(g \mid O, x, I, T, h) = \int p_{\theta_g}(g \mid O, x, I, T, h, \zeta) p_{\theta_{\zeta}}(\zeta \mid O, x, I, T, h) d\zeta. \quad (18)$$

For a discrete intermediate representation, the integral is replaced by a sum. Direct language-conditioned grasp models approximate Equation (1) without exposing ζ , whereas skill-based, grounded, affordance-mediated, and hierarchical methods explicitly or implicitly instantiate the decomposition in Equation (18).

This decomposition can improve interpretability because semantic reasoning and physical grasp generation can be inspected separately. It may also facilitate transfer when ζ is less dependent on a particular robot hand than the joint configuration q . However, the decomposition introduces both an information bottleneck and an error-propagation pathway. If the inferred part, skill, or affordance is incorrect, a downstream grasp generator cannot necessarily recover even when it accurately solves its own conditional subproblem. Conversely, a coarse intermediate representation may omit the orientation, contact, force, or downstream-task information needed to generate a functionally suitable grasp.

4.1. LLM-Based Task Reasoning

Large language models provide broad semantic priors for interpreting underspecified manipulation instructions. A command such as “prepare the table” or “hand me the scissors safely” specifies an intended outcome but may omit the required action sequence,

target part, contact restrictions, grasp orientation, and post-grasp use. Language Models as Zero-Shot Planners demonstrated that pretrained language models can generate action sequences for embodied tasks using knowledge acquired during language pretraining [32]. Linguistic probability alone, however, does not indicate whether a proposed action is available to a specific robot or executable in the current environment.

SayCan combines a language model score with affordance values associated with pretrained robot skills [34]. The language score represents the contextual relevance of a skill, whereas the affordance value estimates whether that skill is likely to succeed from the current state. In terms of Equation (18), the selected robot skill constitutes a discrete intermediate representation ζ . This structure provides embodiment grounding at the level of available skills, but its feasible action space remains restricted to the predefined skill library. Moreover, a learned skill-success value does not explicitly verify the finger-level kinematic, reachability, collision, contact, and safety constraints in Equation (3).

Programmatic approaches expose the dependence on external robot capabilities more explicitly. ProgPrompt constructs situated plans using program-like prompts containing available actions, objects, and executable examples [37]. Code as Policies generates executable programs that invoke perception, geometric reasoning, motion-planning, and control interfaces [36]. These methods can represent conditionals, iterations, spatial computations, and reusable procedures more clearly than an unstructured natural-language plan.

Their physical reliability is nevertheless inherited from the supplied application programming interfaces, perception modules, planners, and controllers. Generated code may be syntactically valid while referring to an incorrectly grounded object, calling an unavailable capability, or requesting a motion for which $\mathcal{G}_{\text{adm}}$ is empty. Program generation therefore does not eliminate physical admissibility verification; it relocates that responsibility to the invoked modules.

Feedback-aware language reasoning partially addresses the open-loop nature of task planning. Inner Monologue incorporates scene descriptions, success detection, object observations, and human or environmental feedback into repeated language model reasoning [35]. The system can revise a high-level plan when an action fails or when an observation contradicts a previous assumption. PaLM-E similarly incorporates visual observations, robot states, and continuous sensor inputs into an embodied multimodal language model [38]. These methods use more of O and x than text-only planners, but they remain dependent on external modules for precise object-part grounding, grasp generation, collision checking, and contact-rich execution.

For dexterous grasping, the most useful role of LLM reasoning is to convert an instruction into structured requirements. From “hand me the scissors safely,” a model may infer the target object, the handover task, a prohibited blade region, and the need to leave a suitable region accessible to the receiver. These inferences must be assigned to the correct component of the formulation.

A mandatory prohibition against blade contact should modify C_{safe} and thereby restrict $\mathcal{G}_{\text{adm}}$. The degree to which a candidate corresponds to the referred object part contributes to S_{sem} , whereas preserving a suitable region and orientation for the receiver contributes to S_{func} . Treating all inferred requirements as compensable language scores would permit strong performance in another objective to offset an unsafe contact.

LLM-based reasoning is strong in semantic breadth, compositionality, and interpretation of incomplete instructions. Its principal limitations include hallucinated skills, incorrect task decomposition, sensitivity to prompts and action descriptions, dependence on external embodiment interfaces, and difficulty distinguishing prohibitions from preferences. LLM output should therefore be interpreted as task structure, constraints, and

conditioning information for Equation (1), rather than as direct evidence that Equation (3) has been satisfied.

4.2. Vision–Language Grounding

Vision–language grounding determines how the instruction I refers to the object or scene representation O and the observation x . In robotic manipulation, grounding may occur at the object, object-part, spatial-relation, affordance, or contact-region level. These levels require different degrees of spatial precision. Object-level grounding may identify which mug is referenced, whereas contact-level grounding must determine which surfaces can be reached by particular fingers without violating collision, stability, or functional requirements.

CLIP introduced transferable image–text representations through large-scale contrastive pretraining [114]. These representations provide broad semantic discrimination but do not directly provide metrically precise contact locations. CLIPort combines semantic “what” features with a spatial “where” pathway for language-conditioned pick-and-place [115]. This structured representation improves spatial action prediction, but its output remains designed primarily for pick-and-place primitives with simple grippers rather than the multi-contact variable c in Equation (2).

Perceiver-Actor represents RGB-D observations in a voxelized three-dimensional space and predicts discretized six-degree-of-freedom actions conditioned on language [116]. The voxel representation retains more metric and geometric information than global image–text similarity. Nevertheless, an end-effector target specifies neither the complete finger-joint configuration q nor the multiple hand–object contacts represented by c . It therefore provides a spatial action prior rather than a complete dexterous grasprepresentation.

VoxPoser uses language and vision–language models to construct composable three-dimensional value maps for target, affordance, and avoidance regions [117]. This structure maps naturally to the formulation in Section 3. A preferred region can contribute to S_{sem} or S_{func} , whereas a mandatory avoidance region should be transferred to C_{col} or C_{safe} . The resulting motion still depends on the accuracy of scene reconstruction and the downstream motion planner. A region with high semantic value does not necessarily support a reachable, collision-free, and stable multi-finger contact configuration.

Open-vocabulary detectors and segmentation models extend grounding to novel object categories and referring expressions. GLIP, MDETR, OWL-ViT, and Grounding DINO associate linguistic expressions with visual regions [100–102,118], while Segment Anything produces prompt-conditioned segmentation masks [119]. These models can supply candidate regions for semantic evaluation, functional evaluation, or safety constraints, but they do not independently determine grasp admissibility.

A segmented handle may contain occluded, unreachable, or geometrically unsuitable surfaces. A mask can leak into a prohibited region or omit a narrow contact area needed for a stable grasp. Detection accuracy and mask overlap therefore do not directly measure whether a resulting candidate satisfies C_{reach} , C_{col} , or C_{contact} .

The mathematical interface between grounding and grasp generation is represented by the intermediate variable ζ in Equation (18). A grounding model may estimate a distribution over possible parts or affordances, after which the grasp generator conditions on the selected or sampled representation. Using only the maximum-probability region discards grounding uncertainty and can make the overall system overconfident. Propagating several grounding hypotheses preserves ambiguity but increases the number of grasp candidates and the associated verification and evaluation cost.

Representative failure modes include referring expression ambiguity, object-part confusion, depth and scale error, occlusion, segmentation leakage, and mismatch between coarse

semantic regions and fine contact geometry. Section 5 examines how dexterous-grasping methods convert these grounded representations into T_h , q , c and, when available, f .

4.3. Vision–Language–Action Policies

Vision–language–action policies reduce the separation between semantic reasoning and robot control by generating actions directly from language, visual observations, and robot states. Their relationship to Equation (1) depends on whether the action is itself a grasp candidate, a trajectory toward a grasp, or an arm-level command that must subsequently be converted into g .

RT-1 demonstrated large-scale language-conditioned robot control using a transformer trained on real-robot trajectories [39]. RT-2 extended this approach through co-training on robot trajectories and web-scale vision–language data and represented robot actions in a token-compatible format [40]. This enables semantic concepts acquired from non-robotic data to influence action generation. Tokenization also permits the use of autoregressive architectures, although discretization can reduce continuous-control precision and autoregressive execution can accumulate errors over time.

Open X-Embodiment aggregates data across different robots, tasks, and institutions [41]. Cross-embodiment data improve task coverage but introduce heterogeneity in action dimensions, kinematics, controller conventions, gripper mechanisms, sensing, and calibration. Octo uses a transformer-based generalist policy and diffusion-style action generation to support heterogeneous observation and action spaces [42]. OpenVLA adapts a pretrained vision–language backbone to robot control and supports parameter-efficient fine-tuning [43]. These approaches reduce the need to train each policy from scratch, but a compatible robot action interface and embodiment-specific adaptation data remain necessary.

Recent systems investigate spatially richer and continuous action representations. 3D-VLA and SpatialVLA incorporate explicit three-dimensional or spatial structure into vision–language–action modeling [44,47]. The π_0 model uses a flow-matching action expert for continuous robot control [45], while RDT-1B employs diffusion modeling for multimodal bimanual action distributions [46]. Continuous generative actions can represent several successful trajectories for the same instruction without independently discretizing each control variable. Their trade-offs include increased training and decoding costs, closed-loop replanning requirements, and the need to maintain consistency between a generated action chunk and a changing contact state.

Across these models, physical admissibility is generally learned implicitly from demonstrations or delegated to a downstream controller, collision checker, or safety layer. This differs from explicitly verifying Equation (3). A low empirical collision frequency within the training distribution is not equivalent to a guarantee that $C_{\text{col}}(g; O) \leq 0$ or $C_{\text{safe}}(g; I, T, h) \leq 0$ under unseen geometry, perception error, or accumulated action drift.

A learned or VLA-generated candidate $\hat{g}$ can conceptually be connected to the Section 3 formulation through constrained projection:

$$g_{\text{proj}} \in \underset{g \in \mathcal{G}_{\text{adm}}}{\operatorname{argmin}} (d_g(g, \hat{g})). \quad (19)$$

Here, d_g denotes a task-dependent discrepancy between two grasp candidates. It may include differences in hand pose, finger-joint configuration, contact region, force profile, or functional intent. Equation (19) provides a common interpretation of methods that refine, project, or filter a learned proposal.

Because $\mathcal{G}_{\text{adm}}$ is generally non-convex, the projection may be non-unique, computationally difficult, or sensitive to the definition and numerical scale of d_g . If $\mathcal{G}_{\text{adm}}$ is empty,

Equation (19) has no admissible solution. Moreover, a projection minimizing geometric discrepancy may fail to preserve semantic intent or downstream functionality. The discrepancy definition cannot therefore be selected independently of S_{sem} and S_{func} .

The principal VLA trade-offs concern scale, representation, and control. Large datasets and models improve semantic and task coverage but increase data-curation, hardware, and inference costs. Action tokens provide architectural compatibility with language models but can lose continuous precision. Diffusion- and flow-based actions represent multimodality but increase decoding and closed-loop integration requirements. None of these representational choices alone guarantees contact-level admissibility.

4.4. Challenges in Extending VLA Models to Dexterous Hands

The principal obstacle to extending general VLA models to dexterous hands is not merely a larger number of action dimensions. The output representation, physical constraints, data distribution, and evaluation criteria all differ from those of arm-centric or parallel-gripper manipulation.

First, most general VLA outputs do not cover all components of Equation (2). An end-effector pose and gripper state may approximate T_h , but they do not specify a high-dimensional q , multiple contacts c , or force- and compliance-related variables f . Even a joint-level policy may leave the contact structure implicit. A visually plausible hand posture may therefore fail to specify which finger links should contact which object surfaces or how the grasp should resist disturbance.

Second, dexterous admissibility is restrictive and discontinuous. Finger contacts create multiple modes, and small changes in T_h or q can cause discontinuous changes in c . Joint limits, self-collision, object penetration, reachability, friction, actuator limits, and mandatory safety requirements jointly determine membership in $\mathcal{G}_{\text{adm}}$. Demonstration likelihood does not guarantee membership in this non-convex set.

Third, general robot datasets are dominated by arm motion and simple-gripper operation. Dexterous learning additionally requires finger-level states, hand-specific kinematics, contact information, synchronized tactile or force observations, and reliable object poses. Robotic hands differ in finger number, joint topology, actuation, kinematic coupling, finger-tip geometry, workspace, and sensing. Cross-embodiment transfer therefore requires more than action normalization or rescaling.

Fourth, conventional task success does not distinguish the components of Equation (7). A robot may lift the correct object while using an instruction-inconsistent contact region, blocking the intended downstream function, violating a user preference, or incurring high execution risk. Physical, semantic, functional, user-dependent, and risk-related outcomes must therefore be reported separately.

Table 5 compares representative language-guided manipulation models and their roles in the common formulation.

Table 5. Representative language-guided robotic manipulation models and their relation to the common formulation.

Study	Methodological Role and Primary Objective	Input Modality	Coverage in the Common Formulation	Limitation for Dexterous Grasping
Code as Policies [36]	Programmatic policy generation from natural-language instructions and available robot interfaces	Natural-language prompt, API descriptions, and object or scene information	Generates an intermediate program ζ that specifies robot behavior. Physical admissibility depends on the downstream perception, planning, and control modules rather than being explicitly represented in the formulation.	Syntactically valid programs may still request infeasible motions or unsupported dexterous-hand behavior.

Table 5. Cont.

Study	Methodological Role and Primary Objective	Input Modality	Coverage in the Common Formulation	Limitation for Dexterous Grasping
VoxPoser [117]	Language-conditioned planning using composable three-dimensional value maps and spatial constraints	Language, visual observation, robot state, and vision–language grounding	Represents semantic and functional preferences through three-dimensional value maps. Preferred regions contribute to S_{sem} and S_{func} , whereas prohibited regions can be incorporated into C_{col} and C_{safe} .	High-value spatial regions do not necessarily admit reachable and stable multi-finger contacts.
VIMA [120]	Multimodal prompt-conditioned manipulation intended to generalize across task specifications	Text prompts, visual prompts, and object observations	Uses multimodal task information to condition action generation in Equation (1). Grasp contacts and force interactions are represented only implicitly.	Remains primarily arm-centric and is not specialized for high-dimensional dexterous contact control.
RT-2 [40]	Vision–language–action policy that transfers web-scale semantic representations to robot control	Image observations, language instructions, and robot trajectory data	Produces language-conditioned robot actions learned from demonstrations without explicitly enforcing the admissibility constraints in Equation (3) or modeling contact states and forces.	Mostly evaluated with relatively simple end effectors and does not explicitly model c or f .
Octo [42]	Open-source generalist policy trained on heterogeneous multi-task robot demonstrations	Image observations, language instruction, robot state, and embodiment-specific action data	Approximates the conditional manipulation policy in Equation (1) while supporting embodiment adaptation. Physical admissibility remains dependent on training data and downstream controllers.	Requires substantial adaptation for dexterous joint spaces, contact-rich grasping, and coordinated finger control.
OpenVLA [43]	Generalist vision–language–action model adapted from a pretrained vision–language backbone	Image observations and language instructions	Produces language-conditioned robot actions but generally represents arm-level motion and simple gripper commands rather than the complete dexterous grasp defined in Equation (2).	Requires embodiment-specific fine-tuning together with explicit contact-admissibility and safety verification for dexterous hands.

Overall, these methods provide semantic structure, grounding, or action priors for Equation (1), but they do not eliminate the need to represent the complete dexterous grasp through Equation (2), enforce Equation (3), and evaluate the objectives in Equation (7).

5. Dexterous Grasping

Dexterous grasping concerns how the candidate variable g in Equation (1) is represented, generated, refined, and executed. Equation (2) supplies a common superset for comparing output representations, whereas Equation (3) distinguishes a generated candidate from an admissible grasp. Equation (7) then defines the objective components used to compare admissible candidates.

The reviewed methods include contact- and configuration-centered representations, optimization-based planning, probabilistic generation, functional and language-guided prediction, tactile adaptation, and dataset or evaluation design. These categories overlap. A diffusion generator may use contact maps, physical refinement, language conditioning, and tactile feedback within one system. The purpose of this classification is therefore to expose methodological assumptions and interfaces rather than to assign each study to a single exclusive category.

5.1. Contact and Configuration Representations

Dexterous-grasping methods differ in which components of Equation (2) they represent explicitly and which components are recovered through optimization, control, or physical interaction.

Configuration-centered methods directly generate or optimize the hand pose T_h and joint configuration q . Their outputs can be mapped to robot kinematics and controllers, making them suitable as terminal targets for grasp execution. However, the resulting contacts can remain implicit until the hand is placed near the object. Similar values of T_h and q can also produce different contacts when object geometry, pose, compliance, or controller tracking changes.

Contact-centered methods instead represent contact points, contact regions, contact maps, surface correspondences, or hand–object distance fields. These representations state more explicitly where the hand should interact with the object, but an additional inverse-kinematics, optimization, or learned-decoding stage is required to recover compatible values of T_h and q . A desired contact pattern does not necessarily admit a physically admissible realization for a particular robotic hand.

Classical analytic grasping represents contact through points, surface normals, friction cones, grasp matrices, and object wrenches [1–3]. These representations provide interpretable foundations for C_{contact} and S_{phys} . Force closure determines whether the grasp can resist arbitrary external wrenches under the assumed contact model, while grasp wrench space measures compare the margin with which different disturbances can be resisted.

The validity of these analytical measures depends on physical assumptions. Common assumptions include rigid object geometry, known contact locations, Coulomb friction, point or soft-finger contact models, and bounded contact forces. These assumptions become less accurate for compliant fingertips, deformable objects, uncertain friction, distributed surface contacts, and sensor-dependent contact estimation. Consequently, a high analytical grasp-quality score does not always predict successful real execution.

Simulation-based datasets and learned models employ more flexible configuration representations. DexGraspNet generates a large number of simulated multi-fingered configurations and evaluates their geometric and physical plausibility [18]. UniDexGrasp separates diverse terminal-grasp proposal generation from the goal-conditioned policy used to reach the selected target [19]. Dexterous Grasp Transformer directly predicts high-dimensional hand configurations from object observations [25].

These approaches improve candidate-generation scalability, but their outputs inherit the assumptions used to construct their training data. Relevant factors include the object distribution, hand model, collision tolerance, contact parameters, grasp-quality threshold, and simulator configuration. A candidate considered admissible during data generation may become inadmissible after transfer to another hand, object representation, controller, or physical environment.

Human hand–object datasets provide additional posture and interaction priors. DexYCB and HO-3D contain human hand and object poses acquired from real interactions [121,122]. OakInk and AffordPose additionally provide intention-, affordance-, or interaction-related annotations [123,124]. Such data can support the learning of natural contact patterns and functionally meaningful grasp configurations.

However, human and robotic hands differ in morphology, joint topology, joint limits, actuation, fingertip geometry, compliance, and sensing. Human-to-robot retargeting may preserve fingertip positions, joint similarity, contact regions, palm orientation, or task function, but these objectives are not equivalent. A configuration that closely preserves human joint geometry may violate robot joint limits, whereas one that preserves fingertip positions may alter the palm orientation or internal-force capability.

Retargeted configurations must therefore be evaluated using Equation (3). Similarity to a human demonstration is not sufficient evidence that a robot grasp belongs to $\mathcal{G}_{\text{adm}}$.

For language-guided grasping, representation quality also depends on S_{sem} and S_{func} . Two object regions can have similar local geometry but different semantic or functional roles.

Conversely, an object part identified as semantically desirable may be too small, occluded, or unreachable for the current hand. Pose reconstruction error and contact overlap should therefore be reported together with physical admissibility and downstream-task outcomes.

5.2. Optimization-Based Dexterous Grasp Planning

Optimization-based dexterous grasp planning searches over one or more components of g while imposing or approximating the constraints in Equation (3). Its principal mathematical role is to construct, explore, or locally refine $\mathcal{G}_{\text{adm}}$ and to improve S_{phys} among the candidates that satisfy the mandatory requirements.

Typical physical terms include contact distance, object penetration, self-collision, joint-limit margins, reachability, posture regularization, force closure, contact-force feasibility, and grasp-wrench-space quality [1–4]. The distinction between hard constraints and soft objectives introduced in Section 3 is essential when interpreting these terms.

Joint-limit violation, severe object penetration, unreachable wrist configurations, and mandatory contact prohibitions should normally define $\mathcal{G}_{\text{adm}}$. In contrast, larger clearance, more natural posture, improved contact distribution, and greater disturbance resistance can be used to rank admissible candidates through S_{phys} .

Replacing every hard requirement with a finite penalty does not generally guarantee admissibility. Suppose that a collision penalty is combined with a large positive physical-quality term. A candidate with a small collision violation may still obtain a higher composite value than a fully admissible candidate if the quality improvement is sufficiently large. The outcome also depends on the scale assigned to each term. Constraint residuals should therefore be reported separately, even when a weighted penalty objective is used internally.

Optimization provides several advantages. Individual physical assumptions can be inspected, hand-specific limits can be imposed explicitly, and a candidate generated by another model can be refined for the current object and embodiment. However, the resulting problems are generally non-convex because finger kinematics, collision geometry, friction constraints, and contact modes are nonlinear.

Different contact combinations can form disconnected feasible regions. The active constraint structure changes when a finger establishes, loses, or slides through a contact. Optimization can therefore be sensitive to initialization, object-pose error, friction assumptions, contact hypotheses, and numerical tolerances.

Global, sampling-based, and multi-start optimization increase contact-mode coverage but require greater computation. Local gradient-based refinement is more efficient but can converge to a locally admissible configuration with poor semantic or functional suitability. Equation (19) illustrates this issue by interpreting refinement as a projection of a generated candidate $\hat{g}$ onto $\mathcal{G}_{\text{adm}}$. Because the admissible set is non-convex, this projection can be non-unique and can alter the original grasp intent.

Learning-based systems do not remove the need for physical optimization. DexGrasp-Net uses optimization and simulation to construct large-scale grasp data [18]. UniDexGrasp learns proposal generation and goal-conditioned execution from physically generated targets [19]. DexGrasp Anything incorporates physics-aware guidance into grasp generation and refinement [27]. Learned proposal distributions and explicit physical verification therefore address complementary parts of the problem.

In a language-guided pipeline, grounding or affordance prediction can first identify a desired or prohibited object region. Optimization can then modify T_h and q while preserving the intended region and satisfying Equation (3). This decomposition is useful when semantic requirements can be converted into geometric constraints or differentiable objectives.

The decomposition becomes less reliable when S_{func} depends on a downstream interaction that cannot be inferred from a static grasp or when S_{user} is defined through sparse

and non-differentiable feedback. In such cases, physical optimization must be combined with task execution, learned evaluation, or human feedback.

5.3. Probabilistic and Generative Grasp Models

Equation (1) captures the motivation for probabilistic grasp generation. The same object and observation can support several physically admissible grasps, and the appropriate distribution changes with the instruction I , downstream task T , feedback h , and robot-hand embodiment. A deterministic predictor can collapse to a frequent configuration even when several distinct contact modes are valid.

DexGraspNet 2.0 learns generative dexterous grasping in large-scale synthetic cluttered scenes [24]. DexDiffuser uses diffusion modeling to sample diverse dexterous grasp candidates and combines generation with evaluation or refinement [26]. UGG develops a unified generative formulation for grasping [28]. DexGrasp Anything incorporates physical awareness into generation and refinement [27]. The preprint-only AnyDexGrasp study [29] and RobustDexGrasp [30] further investigate efficient transfer across hands and robustness to observation or execution variation.

A useful property of a generative model is the probability mass that it assigns to the admissible grasp set:

$$\rho_{\text{adm}} := \Pr_{g \sim p_{\theta}(\cdot | O, x, I, T, h)} [g \in \mathcal{G}_{\text{adm}}]. \quad (20)$$

The admissible probability mass ρ_{adm} distinguishes distributional diversity from physical and safety admissibility. A model may generate many distinct contact modes while assigning substantial probability to collision, penetration, joint-limit violation, or unstable-contact regions. Conversely, a conservative generator may obtain a high ρ_{adm} by repeatedly producing a narrow family of robust power grasps. Admissibility and diversity should therefore be evaluated separately.

In experiments, ρ_{adm} can be operationalized as the observed proportion of generated candidates that satisfy all constraints defining $\mathcal{G}_{\text{adm}}$. This proportion depends on both the generator and the validator. A high value can be misleading if collision, contact stability, or safety requirements are checked inaccurately.

The admissible candidates may subsequently be processed through the Pareto, weighted, or risk-sensitive selection rules in Equations (8)–(12). This separates three questions: whether the model generates diverse candidates, whether those candidates satisfy mandatory constraints, and which admissible candidate provides the desired trade-off among the objectives in Equation (7).

The principal generative trade-off is coverage versus downstream candidate-assessment cost. Sampling more candidates increases the probability of covering disconnected contact modes and finding a candidate with high semantic or functional value. It also increases collision checking, physical evaluation, motion planning, and simulation cost.

Additional diffusion denoising or physical-refinement steps may improve candidate quality while reducing responsiveness. Strong physical guidance may increase the proportion of admissible candidates but bias the generator toward contact modes that are frequent in the training data or easy to evaluate using the assumed physical model. Rare but valid functional grasps may therefore be suppressed.

Cross-hand generalization creates an additional representation problem. A probability distribution over the joint space q of one hand is not directly defined for another hand with a different number of joints or kinematic topology. Hand-conditioned decoders preserve direct executability but require embodiment-specific data. Contact- or affordance-based intermediate representations may transfer more readily but require a reliable procedure for recovering embodiment-specific values of T_h and q .

Evaluation of generative grasping should report the proportion of generated candidates satisfying Equation (3), diversity across distinct contact modes, number of generated candidates, generation time, validation time, average candidate quality, and selected-candidate performance. Reporting only the best result obtained from a large candidate set can conceal low sampling efficiency and poor calibration of the underlying distribution.

5.4. Functional and Language-Guided Dexterous Grasping

Functional grasping extends physical grasp synthesis by considering the downstream consequences of a grasp. Semantic grasping, affordance detection, and task-oriented grasping established that preferred contact regions depend on object function and intended use [55–58]. These studies motivate the distinction between S_{sem} and S_{func} in Equation (7).

Under the evidence definition in Section 2, the five studies reviewed in this subsection [50–54] constitute the direct-evidence subset of the corpus because they explicitly connect language or language-conditioned multimodal input with dexterous or multi-fingered robotic grasp generation or execution. This evidence classification describes the relationship of each study to the focal review problem rather than its methodological quality.

A grasp may be semantically aligned but functionally unsuitable. For example, it may contact the object part referred to in the instruction while orienting the wrist in a way that prevents subsequent tool use. Conversely, a physically and functionally effective grasp may contradict an explicit instruction regarding which region should be contacted or left accessible.

Neither semantic alignment nor functional suitability replaces the minimum physical and safety requirements in Equation (3). A model may correctly identify a handle while producing an unreachable or penetrating configuration. A high semantic score should not compensate for such a violation.

Grasp as You Say directly conditions dexterous grasp generation on fine-grained language guidance [50]. Direct conditioning simplifies the interface between instruction interpretation and grasp output but requires paired data covering linguistic variation, object parts, grasp types, hand configurations, and task contexts. The model must learn semantic interpretation and high-dimensional physical structure jointly.

Multi-GraspLLM uses multimodal language reasoning for semantic grasp generation across multiple robotic hands [51]. Its multi-hand setting addresses embodiment diversity, but a common semantic representation must still be mapped to each hand's joint limits, reachable contacts, and force capability. Common language conditioning does not make the corresponding joint spaces interchangeable.

AffordDexGrasp introduces a generalizable instructive affordance representation between high-level language and low-level grasp generation [52]. This structure corresponds to the intermediate representation ζ introduced in Equation (18). The intermediate affordance improves interpretability by exposing which object region or function is inferred before the final grasp is generated.

The same decomposition creates an error-propagation pathway. An incorrect or incomplete affordance representation restricts the downstream generator to an inappropriate portion of the grasp space. A physically accurate generator cannot recover the correct grasp when its semantic condition is wrong.

DexVLG scales vision–language–grasp prediction using visual, linguistic, and simulated grasp data [53]. It approximates the conditional distribution in Equation (1) using a single-view RGB-D observation and language instruction. Its physical admissibility nevertheless remains dependent on the grasp-generation and physical-verification procedures used to construct its training data, while occluded geometry remains uncertain.

DexGraspVLA separates high-level vision–language reasoning from low-level dexterous execution [54]. This hierarchical organization reduces the burden on one model to perform semantic reasoning and contact-rich control simultaneously. Its effectiveness depends on whether the high-level representation preserves sufficient information about the target part, approach direction, contact structure, safety requirements, and downstream use.

These methods can be grouped into three broad strategies: direct language-to-grasp conditioning, part- or affordance-mediated grasp generation, and hierarchical reasoning and dexterous control. Direct conditioning reduces pipeline complexity but requires extensive paired language–grasp data. Intermediate representations improve interpretability and potential transfer but introduce information bottlenecks. Hierarchical policies support closed-loop execution but depend on the expressiveness and correctness of the interface between levels.

A persistent evaluation gap is that semantic alignment, physical execution, and functional suitability are often assessed separately. Contacting the instructed part does not establish stability, and lifting the object does not establish either instruction compliance or downstream-task completion. Language-guided dexterous grasping should therefore report admissibility violations under Equation (3), the relevant objective-level outcomes in Equation (7), and real downstream-task performance separately.

5.5. Tactile and Multimodal Sensing

Visual and geometric observations primarily support candidate generation before contact. After contact begins, tactile, force, proprioceptive, and motor-current signals provide information about contact formation, pressure, slip, local deformation, and unexpected object motion.

These signals form part of the multimodal observation x in Equation (1). They may also provide measurements or control signals associated with the force-, pressure-, wrench-, and compliance-related variables represented by f in Equation (2).

Tactile feedback is particularly useful when the nominal physical model underlying Equation (3) is inaccurate. A candidate may satisfy geometric collision and contact checks but fail on the physical robot because friction, compliance, object pose, or actuator behavior differs from the assumed model. Contact sensing can detect this mismatch and support finger adjustment, force redistribution, or grasp rejection.

DexTouch investigates tactilely guided dexterous interaction [98]. SoftGrasp combines multimodal imitation learning with adaptive dexterous grasping [31].

Tactile information can improve S_{phys} and reduce R_{exec} , but it introduces additional sensing and calibration assumptions. Sensors differ in resolution, spatial coverage, sensitivity, hysteresis, compliance, durability, and response frequency. Their signals must be synchronized with vision, joint states, control actions, and object motion.

Instrumented fingertips may also alter the contact geometry and compliance that the model is intended to measure. Dense tactile arrays provide richer contact information but increase wiring, processing, simulation, and maintenance requirements. Sparse force, pressure, or motor-current signals are easier to deploy but may not localize contacts or slip accurately.

Tactile feedback does not independently determine S_{sem} or S_{func} . It may detect that an object is slipping without determining that the robot is covering a functional button, blocking a pouring opening, or occupying a handle intended for a human receiver. Effective language-guided systems therefore require pre-contact semantic and geometric reasoning together with post-contact physical adaptation.

5.6. Datasets and Evaluation Protocols

Dexterous grasp datasets supervise different components of the common formulation and should not be treated as interchangeable. Synthetic robot-grasp datasets, including DexGraspNet and DexGraspNet 2.0, provide large numbers of configurations with geometric or simulation-based physical labels [18,24]. Their strengths include scale, controlled variation, and direct compatibility with robotic-hand models. Their limitations include simulator bias, imperfect contact parameters, restricted object distributions, and limited task or user context.

Human hand–object datasets, including DexYCB [121], the preprint-only HO-3D [122], OakInk [123], and AffordPose [124], provide realistic human posture, object-pose, and affordance-related information. These datasets provide useful priors for natural interaction and functional intent but do not establish robotic admissibility. Retargeted configurations must still satisfy Equation (3).

Language-conditioned grasp datasets supervise S_{sem} and S_{func} by associating instructions, object parts, affordances, or grasp styles with grasp candidates [50,53]. Their limitations include restricted linguistic coverage, annotation ambiguity, and the cost of constructing fine-grained instruction–grasp pairs. A dataset may encode one interpretation of functional suitability while excluding other valid task- or user-dependent grasps.

Evaluation should distinguish candidate-level, simulation-level, real-robot, and functional or human-centered evidence. Because these levels are not interchangeable, studies should report the robot embodiment, input modality, object split, candidate budget and selection rule, verification or validation conditions, and whether downstream execution was evaluated physically. Section 8 develops these requirements into a component-specific benchmark matrix.

Table 6 compares representative dexterous grasp generation and language-guided dexterous-grasping methods with distinct methodological roles.

Table 6. Representative studies on dexterous grasp generation and language-guided dexterous grasping.

Study	Problem Scope and Modeling Strategy	Input or Conditioning Information	Coverage in the Proposed Formulation	Remaining Limitation
UniDexGrasp [19]	Separates diverse grasp-proposal generation from goal-conditioned policy execution across objects and robotic hands	Object representation, hand embodiment, and desired terminal grasp	Primarily represents T_h and q through physically generated grasp proposals followed by learned policy execution that approximates $\mathcal{G}_{\text{adm}}$. Evaluation focuses mainly on physical success and cross-object generalization.	Limited explicit modeling of semantic intent, functional suitability, and user-dependent preference.
DexGraspNet 2.0 [24]	Large-scale generative dexterous grasping in synthetic cluttered scenes	Object geometry, local scene context, and robotic-hand model	Represents multi-fingered T_h and q with simulated contact information. Candidate generation, collision checking, and physical evaluation primarily improve admissibility and emphasize S_{phys} .	Language-conditioned, functional, and user-dependent selection are not primary objectives.
DexGrasp Anything [27]	Universal dexterous grasp generation using physics-aware diffusion and refinement	Object geometry, robotic-hand model, and physics-related guidance	Represents physics-guided T_h and q . Physical constraints are incorporated during generation and refinement to increase the proportion of admissible grasps and improve S_{phys} .	Requires broader integration with human feedback, functional objectives, and calibrated real-world validation.

Table 6. Cont.

Study	Problem Scope and Modeling Strategy	Input or Conditioning Information	Coverage in the Proposed Formulation	Remaining Limitation
Multi-GraspLLM [51] [†]	Multimodal language reasoning for semantic grasp generation across multiple hand embodiments	Object and task information, language description, and robotic-hand embodiment	Represents high-level semantic grasp reasoning mapped to hand-specific configurations. Improves semantic conditioning and cross-hand transfer, while contact feasibility, force capability, and admissibility still require hand-specific verification.	Requires stronger integration with low-level contact control and real-world physical evaluation.
AffordDexGrasp [52]	Open-set language-guided dexterous grasping through an instructive affordance intermediate representation	Object geometry, language instruction, and predicted affordance representation ζ	Represents affordance-conditioned T_h and q . The intermediate affordance contributes to S_{sem} and S_{func} , while downstream grasp generation and refinement approximate $\mathcal{G}_{\text{adm}}$.	Errors in the intermediate affordance representation may propagate to the final grasp and require broader contact-rich validation.

[†] Available only as a preprint at the time of the final literature search.

The comparison shows that recent methods increasingly connect semantic conditioning with high-dimensional grasp generation. Nevertheless, explicit contact and force representations, calibrated execution-risk estimation, user-dependent selection, and comparable real-world evaluation remain incomplete.

6. Human-in-the-Loop Learning

Section 3.5 defined constraint-, target-, ranking-, and policy-level feedback and related each level to the common formulation. This section reviews representative learning methods that implement these mechanisms, with emphasis on their information content, adaptation mode, transferability, and human burden.

6.1. Learning from Demonstration

Learning from demonstration uses human behavior as positive evidence of desired robot behavior [60]. In dexterous grasping, a demonstration may specify a terminal palm pose, finger configuration, contact pattern, approach trajectory, force strategy, or complete manipulation sequence. It therefore supervises different components of Equation (2) depending on how the demonstration is collected.

Behavior cloning directly learns a mapping from observations to demonstrated actions. Its principal assumption is that the state distribution encountered by the learned policy resembles the demonstration distribution. Small prediction errors may move the robot into states absent from the demonstrations, after which errors accumulate.

Dagger addresses this covariate shift by collecting expert actions on states visited by the learned policy and aggregating the resulting data [107]. RoboMimic studies architectural, optimization, and dataset factors affecting offline imitation learning for manipulation [108]. These principles remain relevant to dexterous control, although expert labeling becomes more costly when arm and finger motions must be coordinated.

Demonstrations may be collected directly through robot teleoperation or inferred from human behavior. DAPG combines demonstration data with reinforcement learning for dexterous manipulation [21]. DexMV transfers manipulation information from human videos to robotic dexterous control [109]. DexVIP uses human hand-pose priors extracted from video to support dexterous grasping [110]. Continuous grasp-function learning models variation among human demonstrations [125].

Robot teleoperation supplies embodiment-compatible actions and may include both arm and finger trajectories. Its disadvantages are the need for specialized interfaces, operator expertise, and substantial collection time. Human video is easier to collect at scale but does not directly reveal robot joint commands, contact forces, tactile events, or reliable observations of occluded fingers.

Human-to-robot pose estimation and retargeting introduce additional error. A human demonstration may be semantically or functionally appropriate while remaining kinematically impossible for the target robot. Retargeted candidates must therefore be checked using Equation (3).

A terminal-grasp demonstration and a trajectory demonstration provide different forms of supervision. A terminal demonstration primarily provides target-level information about T_h , q , and potentially c . It does not specify a collision-free approach, the order in which contacts should form, or recovery behavior.

A complete trajectory provides policy-level information but may contain demonstrator-specific motions that are not required for task completion. Direct imitation can consequently reproduce unnecessary or inefficient behavior. When only the terminal grasp is functionally important, the approach may be generated more effectively through motion planning or reinforcement learning.

Force and tactile demonstrations provide information about f and contact transitions. They can indicate how force should be redistributed after initial contact or how slip should be corrected. Such data are more difficult to collect and are highly dependent on the sensing configuration.

Demonstration studies should report which components of g are supervised, whether terminal targets or complete trajectories are provided, whether contact and force signals are observed, how human behavior is retargeted, how failed or corrective examples are represented, and whether execution is imitated, planned, or learned separately. The number of trajectories alone is not an adequate measure of collection cost or supervision quality.

6.2. Interactive and Language-Based Correction

Interactive correction provides supervision after a user observes a proposed or executed behavior. Compared with a complete demonstration, correction can concentrate human effort on a particular failure. The user may correct the selected object, target part, palm approach, finger placement, force, or downstream function.

Language correction is attractive because it avoids direct control of high-dimensional robot joints. LILAC uses online language corrections in a shared-autonomy framework [64]. DROC distills and retrieves reusable manipulation knowledge from language corrections [65]. Yell At Your Robot uses verbal feedback to improve robot behavior during deployment [66].

The mathematical effect of a correction depends on its feedback level. “Do not touch the blade” introduces a mandatory contact prohibition and should update $\mathcal{G}_{adm}$ through Equation (13). “Move closer to the center of the handle” changes a desired contact region or the ranking among admissible candidates. “Use a precision grasp” may modify a target configuration or grasp-type preference. “Hold it more gently” primarily concerns f and R_{exec} .

Treating every correction as a complete replacement of the original instruction would obscure these distinctions. The system must infer both the meaning of the feedback and the component of the formulation that should be modified.

Language correction also introduces grounding and temporal-credit problems. Expressions such as “more,” “the other side,” “not like that,” and “like before” depend on the previous proposal and interaction history. A correction may refer to the current finger configuration, an earlier object-part decision, the approach trajectory, or a future task outcome.

The system must therefore retain sufficient context to determine whether the correction changes the conditioning in Equation (1), a target component in Equation (2), a constraint in Equation (3), or an objective in Equation (7).

Some language corrections can be reused. For example, “do not touch the sharp edge” may generalize across multiple candidates and related tools. Incorrect generalization, however, may impose a restriction outside the intended context. Reusable correction systems should retain information about the object, task, user, and interaction state in which the feedback was given.

Corrective teleoperation and physical intervention provide more direct action information. The user can replace an unsafe motion or guide the robot along a recovery trajectory. These modalities reduce ambiguity in the desired action but require continuous attention and control expertise.

Physical correction also introduces a takeover transition. The user must detect the problem, assume control, and generate a recovery before the state becomes unrecoverable. This burden increases when arm motion and finger contacts must be coordinated simultaneously.

Correction-based methods should report correction or intervention latency, duration of user control, successful recovery rate, repeated-correction frequency, whether feedback modifies only the current episode or the future policy, and whether corrections transfer to new objects, poses, or instructions.

6.3. Preference and Reward Modeling

Preference feedback is useful when several candidates remain in $\mathcal{G}_{\text{adm}}$ but differ in the objectives of Equation (7). A user can compare candidates without specifying an exact joint configuration. This is useful for handover orientation, predictability, comfort, personal safety margin, and task-dependent contact choices that are difficult to encode analytically.

TAMER and Deep TAMER use human evaluative feedback to shape behavior [61,126]. Reinforcement learning from human preferences learns reward models from comparisons [62]. Preference-based reinforcement learning provides broader formulations for ranking-based supervision [63]. TidyBot illustrates personalized manipulation based on user-dependent preferences [103].

Equation (16) already provides the Bradley–Terry preference model used in this review. The relevant analytical questions are therefore what information is included in the context variable z , how comparison pairs are selected, how many comparisons are required, how preference uncertainty is represented, and whether inferred utility transfers across objects, tasks, sessions, and users.

Preference feedback should rank admissible candidates rather than override mandatory constraints. A preferred candidate cannot justify violation of C_{kin} , C_{col} , C_{contact} , or C_{safe} . Learned preference utility contributes primarily to S_{user} after Equation (3) has removed inadmissible candidates.

The Bradley–Terry model estimates relative utility. As shown in Equation (17), adding the same constant to every utility value does not change the resulting pairwise probability. The additive utility offset is therefore not identifiable from pairwise comparisons alone.

The effective utility scale also depends on the comparison distribution, parameterization, and calibration procedure. Before preference utility is combined with S_{phys} , S_{sem} , S_{func} , or R_{exec} , the calibration or utility-mapping procedure should be reported. Otherwise, the numerical meaning of a weighted preference term is unclear.

Preferences may be inconsistent or non-stationary. Different users may prefer different contact regions or handover orientations, and the same user may change preference according to object contents, task urgency, fatigue, perceived risk, or interaction experience. Preference-based systems should report comparison count, inter-user agreement, intra-user

consistency, uncertainty, contextual variables, and performance for users or tasks excluded from training.

Approval or rejection provides a lower-bandwidth form of evaluation. It indicates whether one candidate is acceptable but provides little information about how a rejected candidate should be modified. Repeated rejection can support candidate filtering or policy adaptation, but it is less informative than a correction or pairwise comparison.

6.4. Adaptation Loops for Dexterous Grasping

A human-in-the-loop dexterous-grasping system forms a closed adaptation loop. The system processes O , x , I , and T according to Equation (1), generates candidates using the representation in Equation (2), removes candidates violating Equation (3), and evaluates the remaining candidates using Equation (7).

Human feedback can then modify the admissible set, the desired grasp target, the ranking among admissible candidates, or the policy governing future generation and execution.

Figure 4 illustrates how the feedback levels introduced in Sections 3.5 and 6 can be incorporated into a closed adaptation loop. After language grounding, candidate generation, admissibility verification, and multi-objective evaluation, the system may present a selected candidate or simulated outcome to the user. The resulting feedback should be routed according to the source of the observed error. Constraint-level feedback modifies the admissible set, target-level feedback revises a desired grasp or trajectory, ranking-level feedback changes the ordering of admissible candidates, and policy-level feedback modifies future generation or execution. The figure is intended as an organizational schematic; individual methods may implement only a subset of the illustrated components and feedback pathways.

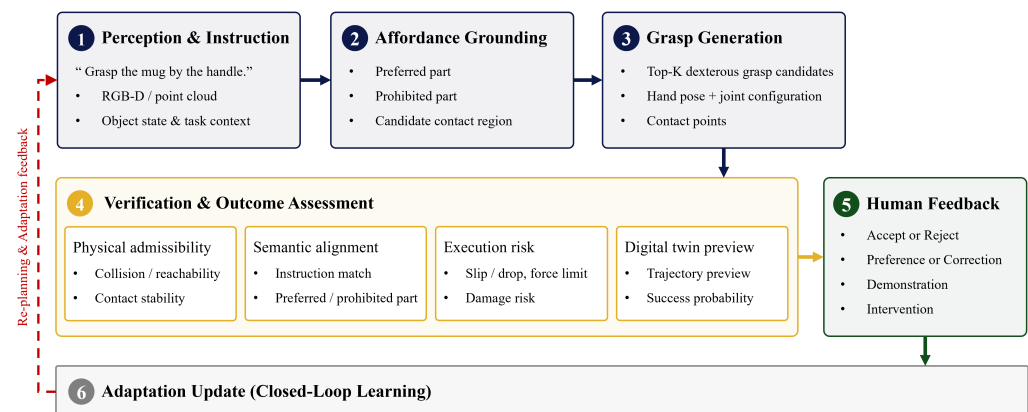


Figure 4. Organizational schematic of a closed-loop human-in-the-loop adaptation process for language-guided dexterous grasping. Perception and instruction processing provide the scene state, task context, and target specification. Affordance grounding identifies preferred, prohibited, and candidate contact regions, after which the grasp-generation module produces candidate hand poses, finger-joint configurations, and contact patterns. The candidates undergo physical-admissibility verification, semantic-alignment evaluation, execution-risk estimation, and digital twin preview before selected outcomes are presented to the user. Human feedback may take the form of acceptance, rejection, preference, correction, demonstration, or intervention and is used to update the admissible set, grasp target, candidate ranking or risk model, and future grasp-generation or execution policy. Individual systems may implement only a subset of the illustrated modules and feedback pathways.

Offline adaptation uses accumulated demonstrations, corrections, and preference labels between deployment sessions. It supports controlled data processing, offline evaluation, and stable training but cannot immediately recover from unexpected deployment failures.

Online adaptation collects feedback during or immediately after execution. It targets states in which the autonomous system is uncertain, unsuccessful, or unsafe. It also introduces non-stationary data, safety concerns, and the possibility that a rapid update degrades behavior that was previously successful.

HIL-SERL combines demonstrations, online human corrections, and reinforcement learning for precise real-world manipulation [69]. Its relevance to this review lies in the integration of deployment-time supervision and policy improvement. It should not be interpreted as evidence that every component of the multi-finger grasp representation in Equation (2) has been explicitly solved.

TRANSIC uses online correction to compensate for errors in a simulation-trained policy [67]. Robot Learning on the Job studies human-assisted autonomy and learning during deployment [68]. These methods demonstrate the value of collecting targeted supervision in states where an autonomous policy fails.

For dexterous grasping, the source of failure should be localized before adaptation. Incorrect object-part grounding should update the grounding module. A semantically correct grasp that slips should update contact regulation, force control, or the physical model. A stable but functionally unsuitable grasp should update task reasoning or S_{func} . A physically successful but disliked grasp should update S_{user} . An approach collision should update the planner, candidate generator, or admissibility test.

Updating the wrong module can increase data requirements without correcting the underlying failure.

Intervention trajectories contain mixed-quality information. The autonomous segment before takeover may consist of actions leading toward failure. The takeover transition may contain hesitation or discontinuity while the user assumes control. The subsequent recovery segment contains the desired corrective behavior.

Treating the entire trajectory as uniformly optimal may reinforce pre-intervention errors and takeover artifacts. Intervention-aware learning should distinguish the autonomous-failure segment, the takeover transition, and the corrective-recovery segment.

Simulation-preview feedback provides a lower-risk interface for candidate-level adaptation. A user can inspect a predicted hand posture, contact regions, collision state, and downstream accessibility before physical execution. The candidate may then be approved, rejected, compared, or revised.

This transfers part of feedback collection from the physical robot to simulation. Its value depends on whether the preview reproduces the failure modes that matter in reality. Section 7 analyzes this dependency through simulation–reality risk discrepancy and false-acceptance rates.

6.5. Feedback Efficiency and Scalability

Human-in-the-loop performance should be evaluated together with human burden. A system that achieves high success through continuous expert correction may be less scalable than a system with slightly lower autonomous performance but substantially lower supervision demand.

A feedback-burden vector can be defined as

$$\left(N_{\text{demo}}, N_{\text{corr}}, N_{\text{pref}}, N_{\text{int}}, T_{\text{human}}, T_{\text{latency}}\right), \quad (21)$$

where N_{demo} , N_{corr} , N_{pref} , and N_{int} denote the numbers of demonstrations, corrections, preference queries, and physical interventions, respectively. T_{human} denotes cumulative human interaction time, and T_{latency} denotes the delay introduced by the interaction loop.

Equation (21) reports feedback burden as a vector because demonstrations, corrections, preference queries, and physical interventions are not directly interchangeable. A scalar

burden may be appropriate for a particular application, but its weights should reflect operator time, expertise, cognitive demand, task interruption, and safety risk.

For a clearly specified performance measure J , with larger values representing better performance, interaction-time efficiency can be reported as

$$\frac{J_{\text{post}} - J_{\text{pre}}}{T_{\text{human}}}, \quad (22)$$

where J_{pre} and J_{post} are measured before and after human feedback.

Equation (22) should be calculated separately for meaningful outcomes such as task success, semantic alignment, downstream-task completion, or a sign-adjusted measure of constraint-violation reduction. It should not be applied to an unexplained composite value of J .

Interactive imitation-learning methods provide mechanisms for allocating a limited feedback budget. HG-Dagger uses human-gated intervention during policy execution [111]. Learning from Interventions interprets takeover as both corrective behavior and negative evidence about the autonomous action [127]. ThriftyDagger uses novelty and risk estimates to determine when an expert should be queried [112].

Feedback modality should be matched to the source of uncertainty. Object ambiguity may require a short language clarification. Uncertain functional suitability may require a pairwise comparison or simulation preview. An unsafe motion may require immediate intervention. An unseen finger-coordination strategy may require a complete demonstration.

Requesting richer feedback than necessary increases burden, whereas requesting insufficient feedback may fail to identify the required correction.

Feedback reuse also differs by level. A constraint such as prohibiting blade contact can apply to many candidates and episodes. A target correction may transfer across similar object poses. User preference may require individual calibration. Policy-level intervention data are frequently coupled to a particular robot, controller, and hand embodiment.

Human feedback is neither noise-free nor stationary. Non-expert users may provide ambiguous or delayed input. Expert demonstrations may include strategies that an autonomous policy cannot reproduce. Surveys of human-in-the-loop reinforcement learning and interactive dexterous imitation learning identify feedback interpretation, burden, safety, and interface design as continuing challenges [88,90].

Representative human-in-the-loop studies are compared in Supplementary Table S5. Across these studies, demonstrations, corrections, preferences, and interventions differ in their information content, adaptation level, sampling distribution, transferability, and human burden.

7. Simulation and Validation Strategies

Simulation and digital twin environments support candidate generation, physical admissibility verification, execution-outcome evaluation, policy training, uncertainty analysis, and pre-execution feedback. Their role should be distinguished from semantic or functional reasoning. A simulator may estimate physical admissibility, physical quality, or execution risk, but it does not independently determine semantic alignment or functional suitability.

Section 7 analyzes simulation through its relationship to Equation (7), S_{phys} , and R_{exec} . A central issue is whether simulated outcomes provide calibrated evidence about real execution. Throughput and visual realism are useful, but neither guarantees predictive validity for contact-rich dexterous manipulation.

7.1. Simulation Platforms

Physics simulators differ in contact models, numerical solvers, robot and object representations, sensor simulation, rendering, parallelism, and integration with learning frameworks. These differences determine which constraints and objectives can be evaluated and at what computational cost.

MuJoCo provides efficient rigid-body dynamics and contact simulation and is widely used for robot control and reinforcement learning [70]. PyBullet provides an accessible physics environment for robotics and learning [71]. Isaac Gym enables GPU-parallel simulation for large-scale robot learning [72]. Isaac Sim supports rendering, sensor simulation, ROS integration, and digital twin-oriented workflows [73]. ORBIT and Isaac Lab extend GPU-accelerated environment construction and training for robot learning [79,128].

Benchmark environments provide standardized scenes, observations, tasks, and evaluation interfaces. RLBench supports vision-based multi-task manipulation [74]. Meta-World supports multi-task and meta-reinforcement learning [75]. Robosuite provides modular MuJoCo-based manipulation environments [76]. SAPIEN focuses on articulated-object interaction [77]. ManiSkill2 provides diverse manipulation tasks and observation modalities [78]. RoboCasa supports large-scale household-task simulation [80].

Platform selection creates a fidelity–throughput trade-off. GPU-parallel simulation supports large-scale sampling over the uncertainty variable ξ and can be used for policy learning, grasp generation, and risk estimation. High visual fidelity supports vision–language perception and user-facing previews but increases rendering and scene-construction cost.

High contact fidelity requires accurate geometry, friction, compliance, actuator behavior, control dynamics, numerical integration, solver configuration, and time discretization. These elements are particularly important for dexterous hands because small errors can change the number, location, or mode of finger contacts.

No simulator name alone guarantees that its contact model reproduces a particular robotic hand, tactile sensor, or object. Simulation studies should report time step, control frequency, contact and solver settings, collision geometry, friction and compliance parameters, actuator models, sensor assumptions, randomization ranges, and computational hardware.

7.2. Simulation-Based Grasp Verification and Validation Evidence

Simulation can support both geometric verification and dynamic validation evidence. Simulation-based verification checks explicit geometric and physical requirements, whereas dynamic simulation evaluates predicted task and execution outcomes under specified simulated conditions. Geometric verification evaluates joint limits, reachability, self-collision, environmental collision, object penetration, and nominal contact placement. It approximates C_{kin} , C_{reach} , and C_{col} , together with geometric aspects of C_{contact} .

Dynamic simulation provides validation evidence concerning lift success, slippage, object displacement, disturbance resistance, contact force, and downstream-task success under the modeled conditions. Because these outcomes depend on the simulator and its parameterization, they do not by themselves validate real-world performance. Dynamic simulation can provide evidence concerning C_{contact} , S_{phys} , and R_{exec} .

Geometric verification is computationally cheaper but cannot determine whether nominal contacts remain stable under dynamics. Dynamic simulation-based validation provides richer outcome evidence but depends on mass, inertia, friction, compliance, actuator behavior, controller frequency, and solver parameters. A geometrically admissible candidate may therefore still fail dynamically.

Using the same uncertainty variable ξ introduced in Section 3, simulated execution risk can be distinguished from real-world execution risk by the distribution from which ξ is drawn. The simulated risk is defined as

$$R_{\text{exec}}^{\text{sim}}(g) = \mathbb{E}\xi \sim P_{\text{sim}}[L(g, \xi)], \quad (23)$$

where P_{sim} denotes the uncertainty distribution represented in simulation and $L(g, \xi)$ denotes the execution loss for grasp g under uncertainty realization ξ .

The uncertainty variable retains the meaning assigned in Section 3. It may include object-pose error, uncertain friction, mass variation, incomplete geometry, perception noise, actuator error, control latency, or another source of execution uncertainty.

The corresponding real-world risk is

$$R_{\text{exec}}^{\text{real}}(g) = \mathbb{E}\xi \sim P_{\text{real}}[L(g, \xi)], \quad (24)$$

where P_{real} denotes the uncertainty distribution encountered during physical deployment.

Equations (23) and (24) use the same uncertainty variable and loss function. Their difference lies in the distribution from which ξ is drawn. In general, P_{sim} and P_{real} need not be identical.

To distinguish the direction of simulation error from its magnitude, the signed risk bias and the absolute risk discrepancy are defined as

$$B_R(g) = R_{\text{exec}}^{\text{sim}}(g) - R_{\text{exec}}^{\text{real}}(g), \quad \Delta R(g) = |B_R(g)|. \quad (25)$$

The signed bias $B_R(g)$ preserves the direction of the simulation error. A positive value indicates that simulation overestimates real-world execution risk, whereas a negative value indicates that simulation underestimates real-world execution risk. The absolute discrepancy $\Delta R(g)$ represents the magnitude of the simulation–reality difference without retaining its direction. A low simulated risk is informative only when $\Delta R(g)$ is sufficiently small and the signed bias does not indicate systematic risk underestimation. Increasing the number of simulation trials can reduce sampling error under P_{sim} , but it cannot remove systematic bias caused by an incorrect contact model, uncertainty distribution, or loss definition.

Simulation may also make false admissibility decisions. Let (g, ξ) denote a grasp candidate and its corresponding operating conditions, sampled from the evaluation distribution P_{eval} . For a constraint C_j included in Equation (3), the constraint-specific false-acceptance rate is defined as

$$\text{FAR}_j = \Pr\left(C_j^{\text{real}}(g, \xi) > 0 \mid C_j^{\text{sim}}(g, \xi) \leq 0\right). \quad (26)$$

Equation (26) represents the probability that a candidate accepted by simulation violates the corresponding constraint during real execution. It can be evaluated separately for collision, contact stability, force limits, or safety constraints.

A simulator with a high internal success rate but a high FAR_j is unsuitable for safety-critical filtering. False acceptance is particularly important because the consequence of accepting an invalid candidate may be more severe than the cost of rejecting a valid one.

Simulation can also support the CVaR-based rule in Equation (12) by estimating losses in the adverse tail of the simulated outcome distribution. Such an estimate is meaningful only when the simulated tail includes the failure modes that occur in reality. Rare physical failures absent from simulation cannot be recovered by drawing additional samples from the same misspecified model.

7.3. Sim-to-Real Transfer

The sim-to-real gap arises because physical dynamics, sensing, control, and uncertainty differ from their simulated counterparts. The gap is particularly severe in dexterous manipulation because small errors in contact position, friction, timing, or force may alter the active contact mode.

Domain randomization exposes perception or control models to varied visual and physical parameters during training [81,129]. Dynamics randomization varies quantities such as mass, friction, damping, actuator strength, control delay, and observation latency [82]. Transferability-aware and adaptive randomization methods update the simulated parameter distribution using transfer evidence or real-world experience [83,84].

Randomization improves robustness when the support of P_{sim} covers relevant real conditions. It may fail when important physical conditions lie outside the randomized ranges, when unrealistic combinations dominate training, or when excessively broad randomization produces unnecessarily conservative behavior. Increasing randomization breadth is therefore not equivalent to reducing Δ_R .

Dexterous in-hand manipulation and robotic Rubik's Cube solving demonstrate that extensive randomization can transfer complex hand policies [22,23]. These results also illustrate the substantial engineering, calibration, data, and computational effort required.

DexPoint studies point-cloud reinforcement learning for sim-to-real dexterous manipulation [20]. Online admittance residual learning provides another mechanism for compensating for contact-rich transfer error [130]. Such approaches reduce the burden on a nominal simulation policy by learning corrective behavior.

Sim-to-real evaluation should report real performance together with the simulated distribution used during training. It should specify which uncertainty sources were randomized, how their ranges were selected, whether real measurements informed the distribution, and which failures arose from perception, geometry, dynamics, contact, or control.

Average success is insufficient when a policy performs well under common conditions but fails severely in a narrow parameter region. Equations (12) and (23)–(25) provide complementary views of expected utility, adverse tail loss, and simulation–reality discrepancy.

Human intervention can support transfer by collecting correction data in real failure states. TRANSIC uses online human correction to compensate for errors in a simulation-trained policy [67]. Such data are informative because they concentrate on states in which the transferred policy is inaccurate or unsafe.

However, intervention data are not sampled from the ordinary deployment distribution. They overrepresent difficult or dangerous states. This bias is useful for learning corrective behavior but should be considered when intervention data are used to estimate general real-world performance.

7.4. Digital Twin-Based Validation

A digital twin differs from a static simulator by maintaining an updated representation of a particular physical robot, object, and environment. It may synchronize robot joint states, object poses, scene geometry, sensor observations, and selected physical parameters.

Digital twin-based validation evidence can therefore be used to assess grasp candidates under conditions intended to correspond to the current deployment state rather than to a generic simulation scene. RialTo reconstructs simulation environments from real observations and uses a real-to-sim-to-real process for robust manipulation [85]. Real-is-Sim investigates dynamic digital twins for robot-policy evaluation [86]. Recent work also considers rapid construction and synchronization of digital twins for safe robotic grasping [87].

Let $s_{\text{real}}(t)$ denote the physical scene state at time t , and let $\hat{s}_{\text{twin}}(t)$ denote the corresponding digital twin estimate. A synchronization error can be defined as

$$e_{\text{twin}}(t) = d_s(s_{\text{real}}(t), \hat{s}_{\text{twin}}(t)), \quad (27)$$

where d_s is a state-discrepancy measure.

Depending on the application, the discrepancy may include robot joint error, object-pose error, geometric reconstruction error, physical parameter mismatch, sensor delay, and synchronization latency. The components and numerical scales included in d_s should be reported explicitly because they are not inherently comparable.

Under the idealized assumption that, for a fixed grasp g , the execution-risk function is locally Lipschitz continuous with respect to the scene state, the risk-estimation discrepancy can be bounded as

$$|R_{\text{exec}}(g; s_{\text{real}}(t)) - R_{\text{exec}}(g; \hat{s}_{\text{twin}}(t))| \leq L_R e_{\text{twin}}(t), \quad (28)$$

where L_R is a local sensitivity constant.

Equation (28) applies only where execution risk varies locally smoothly with scene state. The continuity assumption may fail near contact-mode transitions, collision boundaries, stick-slip transitions, unstable equilibria, and threshold-based safety decisions.

Digital twins can support simulation-preview feedback. Candidate grasps can be visualized, dynamically rolled out, and presented to a user before physical execution. The user may approve, reject, compare, or revise the candidate. This process reduces physical trial risk but introduces perception, synchronization, rendering, simulation, and interaction latency. A visually realistic preview may also cause overtrust when contact and force dynamics are poorly calibrated.

A digital twin should consequently be evaluated through both state accuracy and outcome prediction. Relevant measures include e_{twin} , Δ_R , constraint-specific FAR_{*j*}, synchronization latency, and the frequency with which preview-approved candidates fail during real execution.

7.5. Fidelity, Calibration, and Limitations

Simulation fidelity is multidimensional. Geometric fidelity concerns object shape, environment shape, robot geometry, and pose. Dynamic fidelity concerns mass, inertia, friction, compliance, damping, contact response, actuator behavior, and control. Sensor fidelity concerns image formation, depth noise, tactile response, timing, and latency. Task fidelity concerns whether simulated evaluation reproduces the downstream function and consequences of failure in the physical task.

Improving one dimension does not guarantee improvement in another. A visually photorealistic scene may have inaccurate contact dynamics. A physically accurate rigid-body simulation may omit tactile-sensor behavior. A simulator that predicts lift success may not predict handover safety or downstream tool usability.

Calibration links simulated measurements and parameters to real outcomes. Physical parameters may be estimated through system identification, contact experiments, real rollouts, or adaptive randomization. Closing the Sim-to-Real Loop updates simulation randomization using real-world experience [84]. Real-to-sim-to-real and dynamic digital twin methods similarly update simulated representations using physical observations [85,86].

Calibration should be evaluated on held-out physical executions rather than only on the data used to tune the simulator. Otherwise, a simulator may reproduce known trajectories while remaining inaccurate for new objects, contacts, or task conditions.

Calibration should also be constraint-specific. Accurate reachability prediction does not imply accurate collision, friction, slip, force, or safety prediction. Reporting one aggregate simulation accuracy may conceal severe error in a safety-critical component.

The required fidelity depends on how simulation is used. Approximate physics may be sufficient for broad policy pretraining or coarse rejection of obviously invalid candidates. Candidate ranking requires reliable relative prediction. Risk-sensitive selection requires an informative outcome distribution, including adverse tails. Using simulation for pre-execution safety verification requires validation evidence that false acceptance is sufficiently rare.

Computational cost must also be considered. High-fidelity dynamic rollout for many finger contacts may be too slow for interactive use. Parallel simulation increases throughput but may require simplified contact models or substantial hardware. Digital twin synchronization and human preview introduce additional latency. A practical system must therefore balance candidate count, simulation evaluation depth, uncertainty coverage, response time, and tolerated risk.

Representative simulation, sim-to-real, digital twin, and simulation-preview strategies are compared in Supplementary Table S6. These strategies are not interchangeable: geometric checking, dynamic rollout, randomization, real-world calibration, digital twin synchronization, and human preview provide different forms of verification or validation evidence and should be assessed according to their ability to predict real constraint satisfaction, execution risk, and failure outcomes.

8. Discussion and Future Directions

The preceding sections show that language-guided dexterous grasping cannot be reduced to a single mapping from a natural-language instruction to a robot action. The overall problem contains several distinct decisions: interpreting the instruction, grounding it to objects and object parts, generating high-dimensional grasp candidates, verifying physical and safety admissibility, selecting among competing objectives, adapting to human feedback, and predicting execution outcomes under uncertainty. Each methodological paradigm reviewed in Sections 4–7 addresses only a subset of these decisions.

This section synthesizes the resulting methodological trade-offs rather than repeating the study-level descriptions provided earlier. It first compares the principal paradigms according to their representations, assumptions, computational and data requirements, and characteristic failure modes. It then analyzes how errors propagate across the complete grasping pipeline, identifies limitations in current evaluation practices, discusses the scope and limitations of this review, and formulates actionable research questions for future work.

8.1. Cross-Paradigm Comparison

The reviewed methodologies provide complementary rather than interchangeable capabilities. Language and vision–language–action models provide semantic coverage and task-level generalization, but usually represent contact structure and physical admissibility only implicitly. Optimization-based methods can enforce hand-specific kinematic, collision, contact, and safety requirements, but they depend on initialization, physical assumptions, and computational budget. Generative models can represent multiple grasp modes, but their samples require explicit admissibility verification and selection. Human feedback resolves context- and user-dependent ambiguity, whereas simulation and digital twins reduce physical trial risk only when their predictions are calibrated against real execution.

These differences indicate that the principal question is not which paradigm is universally superior. The more relevant question is which decision each paradigm should control and which assumptions must be verified by another component. A language model may

infer that a blade is a prohibited contact region, but this inference must be converted into a constraint that can be checked against the generated hand configuration. A generative model may efficiently propose diverse candidates, but an explicit validator must determine whether those candidates belong to the admissible grasp set. A digital twin may estimate execution risk, but the estimate must be calibrated against held-out physical trials before it is used for safety-critical acceptance.

Table 7 consolidates these cross-paradigm differences according to each paradigm's principal decision role, assumptions, characteristic failure modes, and appropriate use within an integrated system.

Table 7. Cross-paradigm comparison of methodological approaches for language-guided dexterous grasping.

Methodological Paradigm	Role in the Integrated Framework	Principal Strength	Key Limitation or Failure Mode
Language reasoning and vision–language grounding	Converts instructions into object-, part-, affordance-, constraint-, or intermediate representations for semantic, functional, and safety reasoning	Broad semantic coverage, compositional reasoning, and open-vocabulary interpretation	Ambiguous grounding, hallucinated skills, object-part confusion, or interpreting mandatory prohibitions as soft preferences
Constrained optimization and physics-based refinement	Enforces hand-specific kinematic, reachability, collision, contact, and safety requirements while repairing generated candidates	Explicit and interpretable treatment of physical admissibility	Non-convexity, contact-mode discontinuity, model mismatch, sensitivity to initialization, or loss of semantic intent during refinement
Pareto or weighted multi-objective selection	Ranks admissible candidates according to physical, semantic, functional, user-dependent, and risk-related objectives	Makes trade-offs explicit and supports context-dependent selection	Requires calibrated objective values and justified weights and may produce unstable or compensatory rankings
Probabilistic and generative grasp modeling	Generates diverse high-dimensional grasp candidates across multiple contact modes	Represents multimodal and disconnected grasp distributions	Low admissibility, conservative mode collapse, high sampling cost, or dependence on external verification
Reinforcement learning and VLA-based sequential control	Executes approach motion, contact transitions, recovery, and downstream actions under changing observations	Supports closed-loop and temporally extended manipulation	Compounding action error, sparse-reward ambiguity, embodiment dependence, implicit safety assumptions, or out-of-distribution drift
Human-in-the-loop adaptation	Updates constraints, target variables, candidate rankings, or policy parameters using demonstrations, corrections, preferences, or interventions	Captures task-, context-, and user-dependent requirements unavailable from physical models alone	Human burden, inconsistent feedback, interaction latency, delayed intervention, or feedback applied to the wrong system component
Simulation and digital-twin validation	Filters candidates, estimates execution risk, supports pre-execution preview, and calibrates transfer to physical deployment	Enables repeated low-risk testing and uncertainty analysis	False acceptance, model mismatch, synchronization error, missed rare failures, or user overtrust induced by visually realistic previews

The comparison supports a hybrid interpretation of the field. Semantic models should generate structured conditions rather than be treated as physical validators. Generative methods should provide candidate diversity rather than final safety decisions. Optimization and simulation should verify different aspects of admissibility, while human feedback should be requested selectively where model uncertainty, preference ambiguity, or predicted execution risk justifies the interaction cost.

An integrated system should therefore preserve explicit interfaces between these components. The output of language reasoning should identify whether a requirement is a mandatory constraint, a target, or a preference. The grasp generator should expose the components of the candidate that can be checked or revised. The validator should

return constraint-specific residuals and uncertainty rather than only a binary success label. The selection module should retain separate objective values instead of reporting only a composite score. Finally, the feedback interface should identify which component is being corrected so that adaptation is applied to the relevant module.

8.2. Failure Modes and Safety Considerations

Failures can originate at any stage of the language-to-execution pipeline. Instruction misunderstanding, grounding errors, missing grasp modes, false admissibility decisions, inappropriate objective weights, controller deviation, and simulation-to-reality mismatch require different corrective actions. Therefore, a final task failure should not be treated as a generic policy error. The system should diagnose which component produced the failure before applying feedback or retraining.

At the instruction level, natural-language commands may be ambiguous, incomplete, internally inconsistent, or incompatible with the capabilities of the robot. Commands such as “grasp it safely” do not identify which hazards should be avoided or which object region must remain accessible. Different downstream tasks may also require different grasps even when the literal object reference is unchanged. An instruction can additionally be adversarial or unsafe, intentionally or unintentionally requesting prohibited contact, excessive force, or an action that conflicts with workplace or user-safety constraints.

Language reasoning should therefore determine not only the intended object and task but also the logical status of each requirement. Mandatory prohibitions must be separated from preferences. When the instruction is ambiguous, the system should request clarification rather than commit to an unsupported interpretation. When the instruction conflicts with physical capability or safety policy, the appropriate response is abstention, rejection, or a safe alternative, not optimization of a compromise grasp.

At the grounding level, the correct object may be detected while the relevant part or surface is localized incorrectly. Segmentation leakage can include a prohibited region, and occlusion can remove a narrow but necessary contact surface. Errors in depth, object scale, or pose can cause a semantically correct mask to correspond to an incorrect three-dimensional contact region. Because grounding predictions are often passed to the grasp generator as fixed conditions, early semantic errors can become difficult to recover downstream.

Grounding confidence should therefore be propagated rather than collapsed whenever several interpretations remain plausible. Candidate grasps can be generated under multiple object-part hypotheses, after which physical and task-level evidence can be used to eliminate incompatible interpretations. This increases computational cost but avoids treating an uncertain semantic prediction as a verified physical fact.

At the generation level, a learned model may produce a hand configuration resembling its training distribution while violating the kinematics or geometry of the deployed hand. Common failures include joint-limit violation, self-collision, object penetration, unreachable wrist pose, insufficient contact, and failure to preserve a task-critical region. Generative models may also fail by omitting a rare but necessary contact mode. A high best-of- K success rate can conceal this problem when success is obtained only after generating and validating a large number of poor candidates.

Generation should therefore be evaluated through both coverage and efficiency. The admissible proportion ρ_{adm} , diversity of admissible contact modes, number of candidates required to obtain a successful grasp, and total generation-plus-validation time are all relevant. Reporting only the best candidate does not indicate whether the underlying distribution is reliable enough for interactive or safety-critical use.

At the selection level, failures arise when objectives are improperly scaled, calibrated, or weighted. A physically stable candidate may be selected even though it blocks the intended function, while a semantically preferred candidate may be selected despite a small but unacceptable safety violation. Average utility may also conceal a low-probability failure with severe consequences. The hard-constraint and soft-objective distinction introduced in Section 3 should therefore be preserved throughout selection.

Mandatory physical and safety requirements should remain in the admissible set. They should not be represented only through finite penalties that can be offset by semantic, functional, or preference scores. Among admissible candidates, Pareto and risk-sensitive analysis can expose meaningful trade-offs. Where severe failures are possible, expected utility should be supplemented by adverse-tail measures such as the CVaR term in Equation (12).

At the verification level, simulation may falsely accept a grasp that fails during physical execution. Such false acceptance is more consequential than ordinary numerical prediction error because it permits an invalid candidate to proceed. The constraint-specific false-acceptance rate in Equation (26) should be evaluated separately for collision, contact stability, excessive force, and other safety requirements. Aggregate simulation accuracy can remain high while one safety-critical constraint is predicted poorly.

Digital twin preview introduces an additional human-factors risk. A visually realistic animation may cause the user to assume that contact, friction, and force predictions are also reliable. Preview systems should consequently display uncertainty, constraint residuals, and unvalidated aspects of the model. A user approval obtained from a misleading preview should not be treated as independent evidence of physical safety.

At the execution level, perception delay, controller tracking error, actuator saturation, unexpected compliance, object motion, and contact-mode transitions can invalidate a previously admissible plan. Pre-execution constraint verification and simulation-based validation evidence are therefore insufficient without runtime monitoring. Tactile, force, proprioceptive, and visual monitoring should continue during approach and contact. Safety responses should include slowing, replanning, force reduction, grasp release, or transition to human control when the observed contact state differs from the predicted one.

Human intervention is itself affected by latency and attention. An intervention policy that assumes immediate expert takeover may be unsafe when the operator is distracted or the communication delay is significant. Intervention studies should report detection-to-takeover delay, robot motion during that delay, and the state reached before corrective control becomes effective. The system should retain an autonomous safety layer rather than relying on the human as the only mechanism preventing constraint violation.

Overall, safety should be distributed across the pipeline. Language-level rejection prevents invalid goals, explicit constraints prevent compensable safety violations, risk-sensitive selection accounts for uncertain severe outcomes, calibrated simulation reduces physical trial risk, and contact monitoring detects unmodeled execution errors. No single layer is sufficient, and successful task completion should not be interpreted as evidence that every layer operated correctly.

8.3. Benchmark and Evaluation Gaps

A direct quantitative meta-analysis is currently inappropriate because the reviewed studies use different robotic hands, object sets, task definitions, candidate budgets, validation procedures, simulation platforms, controllers, and success criteria. Combining such heterogeneous success rates into a single ranking would imply comparability that the primary studies do not support. The field instead needs component-specific benchmark protocols that identify which part of the overall pipeline is being evaluated and how the reported outcome should be interpreted.

The four evaluation levels introduced in Section 5—candidate, simulation, real-robot, and functional or human-centered evaluation—provide a starting structure. A benchmark should connect these levels so that a method cannot compensate for poor candidate reliability by generating an unreported number of candidates or compensate for semantic failure by reporting only physical lift success.

Candidate-level evaluation should quantify whether the generator produces executable configurations before dynamic rollout. At minimum, studies should report joint-limit validity, reachability, self-collision and object-penetration rates, contact validity, the admissible proportion ρ_{adm} , and diversity among admissible contact modes. The candidate budget and the rule used to select the final grasp must be fixed or reported because best-of- K performance generally increases with K .

Simulation-level evaluation should test dynamic stability and robustness under a specified uncertainty distribution. Relevant outputs include lift or task success, slip, object displacement, contact force, perturbation resistance, and adverse-tail loss. The simulator, time step, solver, contact parameters, randomization variables, parameter ranges, and number of rollouts should be documented. Without this information, simulated success cannot be reproduced or interpreted as an estimate of real execution risk.

Real-robot evaluation should distinguish execution failure from semantic or functional failure. It should report repeated trials across object instances, poses, initial hand configurations, and environmental variations. Results should include constraint violations, task success, downstream function, recovery events, and the discrepancy between simulated and real outcomes. Confidence intervals or variability across trials and random seeds should accompany mean performance.

Semantic evaluation should test whether the selected object, part, contact region, and grasp mode correspond to the instruction. This can include object- or part-grounding accuracy, contact-region overlap, instruction–grasp compatibility, and human judgment. Functional evaluation should separately measure whether the grasp supports the intended downstream action. A grasp that contacts the instructed handle but prevents the required tool orientation should therefore receive a high semantic score but a low functional score rather than being represented by one ambiguous success label.

Human-centered evaluation should report both improvement and burden. The feedback-burden vector in Equation (21) provides a reporting structure for demonstrations, corrections, preference queries, interventions, cumulative human time, and interaction latency. Improvement per unit of human interaction time, as in Equation (22), should be reported for a clearly defined outcome. User count, prior expertise, within-user repetition, preference agreement, and fatigue effects should also be documented.

Sim-to-real and digital twin evaluation should measure prediction quality rather than only the performance of the final transferred policy. The simulation–reality risk discrepancy, constraint-specific false-acceptance rates, digital twin state error, synchronization latency, and failure rate of preview-approved candidates provide complementary evidence. These quantities should be evaluated on physical executions not used for simulator calibration.

Generalization should be decomposed into separate axes. Unseen object instances, unseen categories, unseen poses, paraphrased instructions, unseen tasks, different users, and new robotic hands represent different forms of generalization. Combining them into one “unseen” split prevents readers from determining what knowledge has actually transferred. Cross-hand evaluation should additionally describe how contact and configuration representations are mapped between embodiments.

Table 8 proposes a minimum benchmark matrix.

A useful benchmark need not require every study to solve the complete problem. A language-grounding study may report only semantic and spatial metrics, while a grasp-

generation study may focus on admissibility and physical quality. However, each study should clearly identify which components are evaluated and which are delegated to external modules. Claims of end-to-end language-guided dexterous grasping require evidence across all applicable levels.

Standardization should also preserve multiple robot embodiments rather than defining one hand as the universal target. A benchmark can provide a common object, task, language, and contact representation while allowing hand-specific decoding and admissibility verification. This design would permit evaluation of whether an intermediate representation transfers across hands and whether the resulting configurations remain physically admissible after embodiment-specific realization.

Table 8. Recommended benchmark dimensions and minimum reporting requirements for language-guided dexterous grasping.

Evaluation Dimension	Core Quantitative Measures	Required Test Variation	Essential Reporting Information
Candidate admissibility	Joint-limit validity, reachability, collision and penetration rates, contact validity, and admissible proportion ρ_{adm}	Object instance, pose, scale, clutter, and robotic-hand embodiment	Number of candidates, grasp representation, validator, thresholds, and final candidate-selection rule
Candidate diversity and efficiency	Diversity among admissible contact modes, best-of-K performance, generation time, validation time, and computational demand	Fixed candidate, computation, and execution-time budgets	Sampling steps, refinement iterations, hardware, and degree of parallelization
Physical quality	Lift success, slip rate, force or wrench margin, perturbation resistance, and object displacement	Friction, mass, object pose, perception error, and actuator variation	Physical model, perturbation distribution, number of trials, and controller configuration
Semantic alignment	Object- and part-grounding accuracy, contact-region agreement, and instruction–grasp compatibility	Novel objects, paraphrased instructions, ambiguous expressions, and prohibited-region instructions	Annotation protocol, language granularity, and agreement among annotators
Functional suitability	Downstream-task completion, accessibility of task-critical regions, and required tool or handover orientation	Tool use, pouring, insertion, assembly, storage, and handover tasks	Complete downstream-task definition and explicit criteria for functional success
Human adaptation	Performance before and after feedback, feedback counts, cumulative human time, latency, consistency, and intervention rate	Multiple users, interaction sessions, objects, and task contexts	User expertise, feedback interface, adaptation cost, and user-specific calibration procedure
Sim-to-real reliability	Simulated and real success, risk discrepancy, constraint-specific false-acceptance rate, and adverse-tail loss	Held-out physical objects, contact conditions, and physical parameters	Calibration data, held-out evaluation data, randomization ranges, and simulator configuration
Digital twin reliability	State discrepancy, synchronization latency, outcome-prediction error, and failure rate after preview approval	Scene changes, moving objects, sensing delay, and contact-mode transitions	Twin update rate, reconstructed state variables, uncertainty display, and human-preview procedure
Overall system safety	Constraint-specific violations, near misses, emergency stops, recovery success, and adverse-tail execution risk	Ambiguous, conflicting, or unsafe instructions and perception or actuation disturbances	Safety rules, rejection criteria, fallback behavior, and runtime monitoring frequency

8.4. Limitations of This Review

This review adopts a structured scoping methodology rather than a systematic meta-analysis. The reviewed field spans language-guided manipulation, dexterous grasping, human-in-the-loop learning, tactile perception, simulation, and digital twins, and relatively few studies address all components simultaneously. The literature set therefore includes both directly relevant language-guided dexterous-grasping studies and constituent studies that provide methods for individual parts of the overall problem.

The final corpus was supported by structured searches in Scopus and IEEE Xplore and by verified backward or forward citation searching. Nevertheless, the review should

not be interpreted as an exhaustive systematic database review. The database search was limited to Scopus and IEEE Xplore and to records published from 2021 to 2026, while earlier eligible studies and other relevant studies were traced through citation searching. Google Scholar was used only during the preliminary exploration of terminology and potentially relevant literature and was not included in the quantitative identification, screening, or inclusion counts.

The resulting taxonomy is also an organizational framework rather than an empirically unique partition of the field. Categories overlap, and one study may simultaneously contribute to semantic grounding, generative grasping, human feedback, and simulation. The taxonomy is intended to expose interfaces and assumptions useful for the common formulation, not to establish that alternative classifications are invalid.

Several included studies are recent preprints because language-guided dexterous grasping and generalist robot policies are rapidly evolving. Their results may change after peer review or subsequent revision. The comparative tables distinguish methodological roles and limitations, but they should not be interpreted as assigning equal evidential strength to peer-reviewed studies and preliminary reports.

A quantitative ranking of methods was not performed because the available results differ substantially in embodiment, dataset, task, candidate budget, simulation conditions, and success criteria. Numerical aggregation under these conditions could imply a degree of comparability that the primary studies do not support. Table 8 is therefore proposed as a prospective reporting standard rather than a retrospective normalization of incompatible experiments. The mathematical expressions provide a comparative analytical framework rather than an implemented algorithm or a guarantee of globally optimal or universally safe grasping.

8.5. Actionable Research Questions

8.5.1. How Can Scalable Language-Conditioned Dexterous Grasp Datasets Be Constructed?

Future datasets should associate language not only with an object label or terminal hand pose but also with the components required to evaluate the complete grasp decision. Relevant annotations include the target object, task-relevant part, preferred contact region, prohibited contact region, downstream task, hand pose, joint configuration, contact structure, approach trajectory, physical outcome, and, when available, tactile or force observations.

The dataset should distinguish mandatory constraints from relative preferences. “Do not touch the blade” and “grasp closer to the center” should not be represented as equivalent text labels because they imply different updates to the admissible set and candidate ranking. Negative examples should include physically invalid, semantically incorrect, and functionally unsuitable grasps, with the failure source labeled separately.

Synthetic generation can provide scale, contact information, and controlled physical variation, while real demonstrations provide sensor characteristics, contact uncertainty, and natural task behavior. A useful dataset should combine these sources without treating simulated labels as ground truth for physical execution. Real trials should be used to calibrate simulator-derived labels and identify failure modes not represented in the synthetic data.

Cross-embodiment data also require a representation that is not tied only to one joint space. Object-centric contacts, affordance regions, palm frames, link-level correspondences, and task constraints may support transfer, but the decoded grasp must still be verified for physical admissibility on each hand. Benchmarks should therefore separate transfer of semantic or contact intent from successful realization on a new embodiment.

Dataset quality should be measured through linguistic diversity, object and task coverage, grasp-mode diversity, annotation agreement, physical-admissibility rate, and coverage of difficult or failure-prone states.

8.5.2. How Should Semantic Alignment and Functional Suitability Be Evaluated?

Semantic alignment and functional suitability should remain separate evaluation targets. Semantic alignment concerns correspondence to the instruction: whether the correct object and part are selected and whether the generated contact pattern follows the expressed requirement. Functional suitability concerns whether the resulting grasp enables the intended downstream task.

Future benchmarks should pair static contact annotations with executable tasks such as handover, pouring, cutting, insertion, assembly, button access, storage, and tool use. This makes it possible to identify cases in which a grasp is semantically plausible but functionally ineffective. Conversely, a functionally successful grasp that violates an explicit instruction should not be considered semantically correct.

Human evaluation may be needed for instructions that admit several valid interpretations. Such evaluation should report annotator agreement and should allow multiple acceptable grasps rather than enforcing one canonical target. Preference data should also be distinguished from objective task requirements because different users may legitimately select different candidates.

Evaluation models for semantic or functional suitability should be calibrated against physical task outcomes and human judgments. A high vision–language similarity score is not sufficient evidence that a grasp preserves the required object affordance or is suitable for handover.

8.5.3. How Can Tactile Sensing Be Integrated with VLA Models?

Language and vision primarily support pre-contact interpretation and planning, whereas tactile and force sensing provide information after contact begins. Future VLA systems should therefore model grasping as a multimodal process whose relevant observations change across execution phases.

One direction is to use language and vision to generate an initial task-conditioned contact hypothesis and to use tactile observations to estimate whether the intended contacts have formed. Tactile signals can then trigger finger adjustment, force redistribution, regrasping, or rejection. This requires a representation linking tactile measurements to finger links, object regions, and the semantic role of the contact.

Tactile integration also raises temporal and embodiment-specific challenges. Tactile arrays, joint states, camera observations, and control actions may have different frequencies and delays. Sensor topology and calibration vary across hands, making direct cross-hand tokenization difficult. Future models should investigate asynchronous multimodal fusion, sensor-conditioned encoders, self-supervised contact representation, and intermediate contact-state representations that can be decoded for different hands.

Simulation can support initial tactile-policy training, but simulated tactile response should be calibrated against real sensors. Evaluation should distinguish contact detection, slip prediction, force regulation, and task recovery rather than reporting a single multimodal success rate.

8.5.4. How Can Human Feedback Be Made Low-Burden, Reliable, and Reusable?

Feedback should be requested according to the source and consequence of uncertainty. Object-reference ambiguity may require one clarification question. Uncertain functional suitability may require a pairwise comparison or preview. An unsafe motion requires

intervention, while a previously unseen coordination strategy may require a complete demonstration. Using the same high-burden feedback modality for all cases is inefficient.

Selective querying should combine model uncertainty, candidate disagreement, expected information gain, and predicted execution risk. However, the query policy should not suppress feedback merely because the model is overconfident. Novelty or out-of-distribution detection should therefore supplement ordinary predictive uncertainty.

Feedback reuse should also match its semantic level. A prohibition can often be reused as a constraint across many candidates. A target correction may transfer across similar poses or objects. A user-specific preference may require an individual model and may change over time. Policy-level demonstrations may support broader generalization but require substantially more data and computation.

Future systems should investigate lightweight adaptation through candidate reranking, external memory, modular adapters, constraint libraries, or local reward-model updates before retraining a complete VLA or dexterous policy. Studies should report not only post-feedback performance but also the number and type of feedback events, human time, latency, retraining cost, and retention of previously learned behavior.

8.5.5. How Should Sim-to-Real Transfer and Digital Twin Reliability Be Benchmarked?

Simulation reliability should be evaluated according to its intended decision role. Approximate simulation may be sufficient for rejecting severe penetration or for broad policy pretraining. Candidate ranking requires reliable relative prediction. Risk-sensitive selection requires an outcome distribution that includes adverse failures. Safety filtering requires evidence that constraint-specific false acceptance is rare.

A benchmark should therefore compare simulated and physical outcomes on matched grasp candidates. It should report the simulation–reality risk discrepancy, false-acceptance rate for each mandatory constraint, adverse-tail loss, and calibration error. Parameter identification and randomization ranges should be derived from measurements where possible and tested on held-out objects and contacts.

Digital twins should additionally report state-estimation error, synchronization delay, update frequency, and sensitivity near contact-mode transitions. Average geometric accuracy is insufficient when a small pose error changes collision status or contact stability discontinuously. Preview-approved grasps should be tracked through physical execution so that the reliability of the complete preview process can be measured.

The benchmark should also evaluate computational latency. A physically accurate rollout that cannot complete within the available decision time may not be useful for interactive grasping. Candidate count, fidelity, rollout duration, hardware demand, and tolerated risk should therefore be reported together.

8.5.6. How Can Safety Be Maintained Under Ambiguous, Conflicting, or Adversarial Instructions?

Language-guided robots require a safety policy that is independent of the linguistic plausibility of a command. An instruction should not be executed solely because a language model assigns it high probability or interprets it confidently. Safety-critical prohibitions, robot capability limits, environmental restrictions, and human-proximity constraints should be verified through dedicated mechanisms.

The system should identify when the admissible grasp set is empty or when the instruction admits several materially different interpretations. In such cases, it should request clarification, propose a safe alternative, change the end effector or approach direction, or reject the task. An unsafe requirement should not be softened automatically in order to force an admissible solution.

Adversarial evaluation should include instructions that obscure prohibited actions, conflict with observed scene conditions, refer to nonexistent objects or capabilities, or attempt to override safety rules. Performance should be measured through unsafe-execution rate, appropriate clarification, correct rejection, and success of fallback behavior.

During execution, the safety layer should monitor perception, trajectory, contact, and force independently of the language model. Human approval may provide useful contextual evidence, but it should not override a detected physical violation. Future work should determine how semantic safety rules can be translated into verifiable spatial, contact, force, and temporal constraints while preserving the ability to abstain when that translation is uncertain.

Taken together, these research questions suggest that future progress is more likely to result from explicitly coordinated modules than from increasing the scale of one model alone. Future systems would therefore need to use language and vision to infer structured intent, generate multiple embodiment-conditioned candidates, enforce mandatory constraints, retain multi-objective trade-offs, estimate tail risk using calibrated simulation, monitor contact through tactile sensing, and request human feedback selectively. The central requirement is not that every component be mathematically or computationally perfect, but that its assumptions, uncertainty, failure modes, and decision authority remain explicit.

9. Conclusions

This review examined language-guided dexterous grasping with human-in-the-loop adaptation through a structured scoping methodology and a formulation-oriented analytical perspective. The reviewed literature was organized around four interconnected directions: language-guided robotic manipulation, dexterous grasp generation, human-in-the-loop learning, and simulation-based validation. The main conclusions can be summarized in three points.

First, language-guided dexterous grasping should be understood as a constrained multi-objective decision problem rather than as a direct language-to-action mapping. A complete system must distinguish grasp candidate generation from physical and safety admissibility and then evaluate admissible candidates according to physical quality, semantic alignment, functional suitability, user-dependent preference, and execution risk. A semantically plausible grasp may remain physically inadmissible, whereas a physically stable grasp may violate an instruction, obstruct a task-critical region, or fail to support the intended downstream task.

Second, the reviewed methodological paradigms provide complementary but incomplete capabilities. Language models, vision–language models, and vision–language–action policies provide semantic reasoning, grounding, and action priors, but they generally lack explicit contact-level guarantees. Optimization and physics-based methods can enforce embodiment-specific kinematic, collision, contact, and safety requirements, but they remain affected by non-convexity, physical-model assumptions, and computational cost. Probabilistic and generative models can represent diverse grasp modes, but their outputs still require admissibility verification and real-world validation of the estimated execution risk. Human feedback can resolve task- and user-dependent ambiguity, but it should be routed according to whether it modifies constraints, grasp targets, candidate ranking, or future policy behavior.

Third, reliable deployment requires validation and evaluation protocols that separate the main components of the problem. Simulation and digital twins can support candidate filtering, uncertainty evaluation, risk-sensitive selection, and pre-execution preview, but their predictions should be calibrated against physical execution rather than treated as independent evidence of safety. Future benchmarks should therefore report candidate

admissibility, component-specific grasp outcomes, human-feedback burden, and sim-to-real reliability separately.

Overall, this review suggests that progress in language-guided dexterous grasping depends on explicitly coordinated systems in which language and vision infer structured task requirements, embodiment-conditioned models generate multiple grasp candidates, physical and safety constraints define admissibility, selection methods preserve trade-offs among objectives, tactile or multimodal sensing supports execution monitoring, and human feedback is requested selectively. This perspective provides a basis for developing systems that are physically admissible, semantically aligned, functionally suitable, user-adaptive, and validated for real-world deployment.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/math14152709/s1>.

Author Contributions: Conceptualization, E.J. and S.-C.L.; methodology, E.J. and S.-C.L.; literature search, E.J.; study screening, E.J. and S.-C.L.; data curation, E.J. and S.-C.L.; formal analysis, E.J.; investigation, E.J.; writing—original draft preparation, E.J.; writing—review and editing, E.J. and S.-C.L.; supervision, S.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00562981, RS-2025-25433409).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available in the Supplementary Materials accompanying this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ferrari, C.; Canny, J. Planning Optimal Grasps. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 1992; Volume 3, pp. 2290–2295. [\[CrossRef\]](#)
2. Bicchi, A. On the closure properties of robotic grasping. *Int. J. Robot. Res.* **1995**, *14*, 319–334. [\[CrossRef\]](#)
3. Shimoga, K.B. Robot grasp synthesis algorithms: A survey. *Int. J. Robot. Res.* **1996**, *15*, 230–266. [\[CrossRef\]](#)
4. Miller, A.T.; Allen, P.K. GraspIt! a versatile simulator for robotic grasping. *IEEE Robot. Autom. Mag.* **2004**, *11*, 110–122. [\[CrossRef\]](#)
5. Sahbani, A.; El-Khoury, S.; Bidaud, P. An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.* **2012**, *60*, 326–336. [\[CrossRef\]](#)
6. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-driven grasp synthesis—A survey. *IEEE Trans. Robot.* **2013**, *30*, 289–309. [\[CrossRef\]](#)
7. Yoshikawa, T. Multifingered Robot Hands: Control for Grasping and Manipulation. *Annu. Rev. Control* **2010**, *34*, 199–208. [\[CrossRef\]](#)
8. Mahler, J.; Pokorny, F.T.; Hou, B.; Roderick, M.; Laskey, M.; Aubry, M.; Kohlhoff, K.; Kröger, T.; Kuffner, J.; Goldberg, K. Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2016; pp. 1957–1964. [\[CrossRef\]](#)
9. Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.; Ojea, J.A.; Goldberg, K. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In *Proceedings of the Robotics: Science and Systems XIII*, Cambridge, MA, USA, 12–16 July 2017. [\[CrossRef\]](#)
10. Mahler, J.; Matl, M.; Liu, X.; Li, A.; Gealy, D.; Goldberg, K. Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2018; pp. 5620–5627. [\[CrossRef\]](#)
11. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **2018**, *37*, 421–436. [\[CrossRef\]](#)

12. Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In *Proceedings of the 2nd Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2018; Volume 87, pp. 651–673.
13. Morrison, D.; Leitner, J.; Corke, P. Closing the Loop for Robotic Grasping: A Real-Time, Generative Grasp Synthesis Approach. In *Proceedings of the Robotics: Science and Systems XIV*, Pittsburgh, PA, USA, 26–30 June 2018. [[CrossRef](#)]
14. Ten Pas, A.; Gualtieri, M.; Saenko, K.; Platt, R. Grasp pose detection in point clouds. *Int. J. Robot. Res.* **2017**, *36*, 1455–1473. [[CrossRef](#)]
15. Liang, H.; Ma, X.; Li, S.; Görner, M.; Tang, S.; Fang, B.; Sun, F.; Zhang, J. PointNetGPD: Detecting grasp configurations from point sets. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2019; pp. 3629–3635. [[CrossRef](#)]
16. Fang, H.S.; Wang, C.; Gou, M.; Lu, C. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11444–11453. [[CrossRef](#)]
17. Sundermeyer, M.; Mousavian, A.; Triebel, R.; Fox, D. Contact-GraspNet: Efficient 6-dof grasp generation in cluttered scenes. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2021; pp. 13438–13444. [[CrossRef](#)]
18. Wang, R.; Zhang, J.; Chen, J.; Xu, Y.; Li, P.; Liu, T.; Wang, H. DexGraspNet: A Large-Scale Robotic Dexterous Grasp Dataset for General Objects Based on Simulation. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2023; pp. 11359–11366. [[CrossRef](#)]
19. Xu, Y.; Wan, W.; Zhang, J.; Liu, H.; Shan, Z.; Shen, H.; Wang, R.; Geng, H.; Weng, Y.; Chen, J.; et al. UniDexGrasp: Universal Robotic Dexterous Grasping via Learning Diverse Proposal Generation and Goal-Conditioned Policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2023; pp. 4737–4746. [[CrossRef](#)]
20. Qin, Y.; Huang, B.; Yin, Z.H.; Su, H.; Wang, X. DexPoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Proceedings of the Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; pp. 594–605.
21. Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of the Robotics: Science and Systems XIV*, Pittsburgh, PA, USA, 26–30 June 2018. [[CrossRef](#)]
22. OpenAI; Andrychowicz, M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; et al. Learning Dexterous In-Hand Manipulation. *Int. J. Robot. Res.* **2020**, *39*, 3–20. [[CrossRef](#)]
23. OpenAI; Akkaya, I.; Andrychowicz, M.; Chociej, M.; Litwin, M.; McGrew, B.; Petron, A.; Paino, A.; Plappert, M.; Powell, G.; et al. Solving Rubik’s Cube with a robot hand. *arXiv* **2019**, arXiv:1910.07113. [[CrossRef](#)]
24. Zhang, J.; Liu, H.; Li, D.; Yu, X.; Geng, H.; Ding, Y.; Chen, J.; Wang, H. DexGraspNet 2.0: Learning Generative Dexterous Grasping in Large-Scale Synthetic Cluttered Scenes. In *Proceedings of the 8th Annual Conference on Robot Learning*, Munich, Germany, 6–9 November 2024.
25. Xu, G.H.; Wei, Y.L.; Zheng, D.; Wu, X.M.; Zheng, W.S. Dexterous grasp transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2024; pp. 17933–17942. [[CrossRef](#)]
26. Weng, Z.; Lu, H.; Kragic, D.; Lundell, J. DexDiffuser: Generating dexterous grasps with diffusion models. *IEEE Robot. Autom. Lett.* **2024**, *9*, 11834–11840. [[CrossRef](#)]
27. Zhong, Y.; Jiang, Q.; Yu, J.; Ma, Y. DexGrasp Anything: Towards universal robotic dexterous grasping with physics awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2025; pp. 22584–22594. [[CrossRef](#)]
28. Lu, J.; Kang, H.; Li, H.; Liu, B.; Yang, Y.; Huang, Q.; Hua, G. UGG: Unified generative grasping. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 414–433. [[CrossRef](#)]
29. Fang, H.S.; Yan, H.; Tang, Z.; Fang, H.; Wang, C.; Lu, C. AnyDexGrasp: General dexterous grasping for different hands with human-level learning efficiency. *arXiv* **2025**, arXiv:2502.16420. [[CrossRef](#)]
30. Zhang, H.; Wu, Z.; Huang, L.; Christen, S.; Song, J. Robust Dexterous Grasping of General Objects. In *Proceedings of the 9th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2025; Volume 305, pp. 3035–3050.
31. Li, Y.; Guo, C.; Ren, J.; Chen, B.; Cheng, C.; Zhang, H.; Lu, H. SoftGrasp: Adaptive grasping for dexterous hand based on multimodal imitation learning. *Biomim. Intell. Robot.* **2025**, *5*, 100217. [[CrossRef](#)]
32. Huang, W.; Abbeel, P.; Pathak, D.; Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *Proceedings of the International Conference on Machine Learning*; PMLR: Norfolk, MA, USA, 2022; pp. 9118–9147.
33. Sharma, P.; Torralba, A.; Andreas, J. Skill induction and planning with latent language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 1713–1726. [[CrossRef](#)]

34. Ichter, B.; Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Proceedings of the 6th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 205, pp. 287–318.
35. Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Proceedings of the 6th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 205, pp. 1769–1782.
36. Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; Zeng, A. Code as policies: Language model programs for embodied control. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2023; pp. 9493–9500. [[CrossRef](#)]
37. Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; Garg, A. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2023; pp. 11523–11530. [[CrossRef](#)]
38. Driess, D.; Xia, F.; Sajjadi, M.S.M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. PaLM-E: An Embodied Multimodal Language Model. In *Proceedings of the 40th International Conference on Machine Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 202, pp. 8469–8488.
39. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. RT-1: Robotics Transformer for Real-World Control at Scale. In *Proceedings of the Robotics: Science and Systems*, Daegu, Republic of Korea, 10–14 July 2023. [[CrossRef](#)]
40. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 229, pp. 2165–2183.
41. Open X-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2024; pp. 6892–6903. [[CrossRef](#)]
42. Octo Model Team; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Kreiman, T.; Xu, C.; et al. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of the Robotics: Science and Systems XX*, Delft, The Netherlands, 13–17 July 2024. [[CrossRef](#)]
43. Kim, M.J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.P.; Sanketi, P.R.; Vuong, Q.; et al. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of the 8th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2025; Volume 270, pp. 2679–2713.
44. Zhen, H.; Qiu, X.; Chen, P.; Yang, J.; Yan, X.; Du, Y.; Hong, Y.; Gan, C. 3D-VLA: A 3D Vision-Language-Action Generative World Model. In *Proceedings of the 41st International Conference on Machine Learning*; PMLR: Norfolk, MA, USA, 2024; Volume 235, pp. 61229–61245.
45. Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.R.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. π_0 : A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of the Robotics: Science and Systems*, Los Angeles, CA, USA, 21–25 June 2025. [[CrossRef](#)]
46. Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; Zhu, J. RDT-1B: A Diffusion Foundation Model for Bimanual Manipulation. In *Proceedings of the Thirteenth International Conference on Learning Representations*, Singapore, 24–28 April 2025.
47. Qu, D.; Song, H.; Chen, Q.; Yao, Y.; Ye, X.; Gu, J.; Wang, Z.; Ding, Y.; Zhao, B.; Wang, D.; et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Models. In *Proceedings of the Robotics: Science and Systems*, Los Angeles, CA, USA, 21–25 June 2025. [[CrossRef](#)]
48. Zhao, W.; Ding, P.; Zhang, M.; Gong, Z.; Bai, S.; Zhao, H.; Wang, D. VLAS: Vision-Language-Action Model with Speech Instructions for Customized Robot Manipulation. In *Proceedings of the Thirteenth International Conference on Learning Representations*, Singapore, 24–28 April 2025.
49. Zhou, Z.; Zhu, Y.; Zhu, M.; Wen, J.; Liu, N.; Xu, Z.; Meng, W.; Peng, Y.; Shen, C.; Feng, F.; et al. ChatVLA: Unified multimodal understanding and robot control with vision-language-action model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 5377–5395. [[CrossRef](#)]
50. Wei, Y.L.; Jiang, J.J.; Xing, C.; Tan, X.T.; Wu, X.M.; Li, H.; Cutkosky, M.; Zheng, W.S. Grasp as you say: Language-guided dexterous grasp generation. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 46881–46907. [[CrossRef](#)]
51. Li, H.; Mao, W.; Deng, W.; Meng, C.; Fan, H.; Wang, T.; Tan, P.; Wang, H.; Deng, X. Multi-GraspLLM: A Multimodal LLM for Multi-Hand Semantic Guided Grasp Generation. *arXiv* **2024**, arXiv:2412.08468. [[CrossRef](#)]
52. Wei, Y.L.; Lin, M.; Lin, Y.; Jiang, J.J.; Wu, X.M.; Zeng, L.A.; Zheng, W.S. AffordDexGrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2025; pp. 11818–11828. [[CrossRef](#)]

53. He, J.; Li, D.; Yu, X.; Qi, Z.; Zhang, W.; Chen, J.; Zhang, Z.; Zhang, Z.; Yi, L.; Wang, H. DexVLG: Dexterous vision-language-grasp model at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2025; pp. 14248–14258. [\[CrossRef\]](#)
54. Zhong, Y.; Huang, X.; Li, R.; Zhang, C.; Liang, Y.; Yang, Y.; Chen, Y. DexGraspVLA: A vision-language-action framework towards general dexterous grasping. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2026; Volume 40, pp. 18836–18844. [\[CrossRef\]](#)
55. Dang, H.; Allen, P.K. Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: New York, NY, USA, 2012; pp. 1311–1317. [\[CrossRef\]](#)
56. Myers, A.; Teo, C.L.; Fermüller, C.; Aloimonos, Y. Affordance detection of tool parts from geometric features. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2015; pp. 1374–1381. [\[CrossRef\]](#)
57. Do, T.T.; Nguyen, A.; Reid, I. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2018; pp. 5882–5889. [\[CrossRef\]](#)
58. Fang, K.; Zhu, Y.; Garg, A.; Kurenkov, A.; Mehta, V.; Li, F.F.; Savarese, S. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *Int. J. Robot. Res.* **2020**, *39*, 202–216. [\[CrossRef\]](#)
59. Mo, K.; Guibas, L.J.; Mukadam, M.; Gupta, A.; Tulsiani, S. Where2Act: From pixels to actions for articulated 3D objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 6813–6823. [\[CrossRef\]](#)
60. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483. [\[CrossRef\]](#)
61. Knox, W.B.; Stone, P. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*; ACM: New York, NY, USA, 2009; pp. 9–16. [\[CrossRef\]](#)
62. Christiano, P.F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; Amodei, D. Deep reinforcement learning from human preferences. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–9.
63. Wirth, C.; Akrou, R.; Neumann, G.; Fürnkranz, J. A survey of preference-based reinforcement learning methods. *J. Mach. Learn. Res.* **2017**, *18*, 4945–4990.
64. Cui, Y.; Karamcheti, S.; Palleti, R.; Shivakumar, N.; Liang, P.; Sadigh, D. No, to the right: Online language corrections for robotic manipulation via shared autonomy. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*; ACM: New York, NY, USA, 2023; pp. 93–101. [\[CrossRef\]](#)
65. Zha, L.; Cui, Y.; Lin, L.H.; Kwon, M.; Arenas, M.G.; Zeng, A.; Xia, F.; Sadigh, D. Distilling and retrieving generalizable knowledge for robot manipulation via language corrections. In *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2024; pp. 15172–15179. [\[CrossRef\]](#)
66. Shi, L.X.; Hu, Z.; Zhao, T.Z.; Sharma, A.; Pertsch, K.; Luo, J.; Levine, S.; Finn, C. Yell At Your Robot: Improving On-the-Fly from Language Corrections. In *Proceedings of the Robotics: Science and Systems XX*, Delft, The Netherlands, 13–17 July 2024. [\[CrossRef\]](#)
67. Jiang, Y.; Wang, C.; Zhang, R.; Wu, J.; Li, F.F. TRANSIC: Sim-to-Real Policy Transfer by Learning from Online Correction. In *Proceedings of the 8th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2025; Volume 270, pp. 1691–1729.
68. Liu, H.; Nasiriany, S.; Zhang, L.; Bao, Z.; Zhu, Y. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. *Int. J. Robot. Res.* **2025**, *44*, 1727–1742. [\[CrossRef\]](#)
69. Luo, J.; Xu, C.; Wu, J.; Levine, S. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Sci. Robot.* **2025**, *10*, eads5033. [\[CrossRef\]](#) [\[PubMed\]](#)
70. Todorov, E.; Erez, T.; Tassa, Y. MuJoCo: A physics engine for model-based control. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: New York, NY, USA, 2012; pp. 5026–5033. [\[CrossRef\]](#)
71. Coumans, E.; Bai, Y. PyBullet: A Python Module for Physics Simulation for Games, Robotics, and Machine Learning. 2016. Available online: <https://docs.google.com/document/d/10sXEhzFRSvFcl3XxNGhND4N2SedqwdAvK3dsihxVUA> (accessed on 14 July 2026).
72. Makoviychuk, V.; Wawrzyniak, L.; Guo, Y.; Lu, M.; Storey, K.; Macklin, M.; Hoeller, D.; Rudin, N.; Allshire, A.; Handa, A.; et al. Isaac Gym: High Performance GPU-Based Physics Simulation for Robot Learning. *arXiv* **2021**, arXiv:2108.10470. [\[CrossRef\]](#)
73. NVIDIA Corporation. NVIDIA Isaac Sim. 2024. Available online: https://developer.nvidia.com/isaac/sim?size=n_6_n&sort-field=featured&sort-direction=desc (accessed on 14 July 2026).
74. James, S.; Ma, Z.; Arrojo, D.R.; Davison, A.J. RL Bench: The robot learning benchmark & learning environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 3019–3026. [\[CrossRef\]](#)
75. Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2020; pp. 1094–1100.

76. Zhu, Y.; Wong, J.; Mandlekar, A.; Martín-Martín, R.; Joshi, A.; Lin, K.; Maddukuri, A.; Nasiriany, S.; Zhu, Y. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv* **2020**, arXiv:2009.12293. [\[CrossRef\]](#)
77. Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; et al. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11097–11107. [\[CrossRef\]](#)
78. Gu, J.; Xiang, F.; Li, X.; Ling, Z.; Liu, X.; Mu, T.; Tang, Y.; Tao, S.; Wei, X.; Yao, Y.; et al. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *Proceedings of the Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 1–5 May 2023.
79. Mittal, M.; Yu, C.; Yu, Q.; Liu, J.; Rudin, N.; Hoeller, D.; Yuan, J.L.; Singh, R.; Guo, Y.; Mazhar, H.; et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3740–3747. [\[CrossRef\]](#)
80. Nasiriany, S.; Maddukuri, A.; Zhang, L.; Parikh, A.; Lo, A.; Joshi, A.; Mandlekar, A.; Zhu, Y. RoboCasa: Large-Scale Simulation of Household Tasks for Generalist Robots. In *Proceedings of the Robotics: Science and Systems XX*, Delft, The Netherlands, 13–17 July 2024. [\[CrossRef\]](#)
81. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: New York, NY, USA, 2017; pp. 23–30. [\[CrossRef\]](#)
82. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2018; pp. 3803–3810. [\[CrossRef\]](#)
83. Muratore, F.; Treede, F.; Gienger, M.; Peters, J. Domain randomization for simulation-based policy optimization with transferability assessment. In *Proceedings of the Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2018; pp. 700–713.
84. Chebotar, Y.; Handa, A.; Makoviychuk, V.; Macklin, M.; Issac, J.; Ratliff, N.; Fox, D. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2019; pp. 8973–8979. [\[CrossRef\]](#)
85. Villasevil, M.T.; Simeonov, A.; Li, Z.; Chan, A.; Chen, T.; Gupta, A.; Agrawal, P. Reconciling Reality through Simulation: A Real-To-Sim-to-Real Approach for Robust Manipulation. In *Proceedings of the Robotics: Science and Systems XX*, Delft, The Netherlands, 13–17 July 2024. [\[CrossRef\]](#)
86. Abou-Chakra, J.; Sun, L.; Rana, K.; May, B.; Schmeckpeper, K.; Suenderhauf, N.; Minniti, M.V.; Herlant, L. Real-is-Sim: Bridging the Sim-to-Real Gap with a Dynamic Digital Twin. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Vienna, Austria, 1–5 June 2026.
87. Huang, R.; Yang, B.; Gui, W.; Morgan, J.; Biyik, E.; Li, J. SyncTwin: Fast Digital Twin Construction and Synchronization for Safe Robotic Grasping. *arXiv* **2026**, arXiv:2601.09920. [\[CrossRef\]](#)
88. Retzlaff, C.O.; Das, S.; Wayllace, C.; Mousavi, P.; Afshari, M.; Yang, T.; Saranti, A.; Angerschmid, A.; Taylor, M.E.; Holzinger, A. Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *J. Artif. Intell. Res.* **2024**, *79*, 359–415. [\[CrossRef\]](#)
89. An, S.; Meng, Z.; Tang, C.; Zhou, Y.; Liu, T.; Ding, F.; Zhang, S.; Mu, Y.; Song, R.; Zhang, W.; et al. Dexterous Manipulation Through Imitation Learning: A Survey. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 1760–1792. [\[CrossRef\]](#)
90. Welte, E.; Rayyes, R. Interactive imitation learning for dexterous robotic manipulation: Challenges and perspectives—A survey. *Front. Robot. AI* **2025**, *12*, 1682437. [\[CrossRef\]](#) [\[PubMed\]](#)
91. Sun, J.; Mao, P.; Kong, L.; Wang, J. A Review of Embodied Grasping. *Sensors* **2025**, *25*, 852. [\[CrossRef\]](#) [\[PubMed\]](#)
92. Zeng, F.; Gan, W.; Huai, Z.; Sun, L.; Chen, H.; Wang, Y.; Liu, N.; Yu, P.S. Large Language Models for Robotics: A Survey. *arXiv* **2023**, arXiv:2311.07226. [\[CrossRef\]](#)
93. Kawaharazuka, K.; Oh, J.; Yamada, J.; Posner, I.; Zhu, Y. Vision-Language-Action Models for Robotics: A Review Towards Real-World Applications. *IEEE Access* **2025**, *13*, 162467–162504. [\[CrossRef\]](#)
94. Wolf, R.; Shi, Y.; Liu, S.; Rayyes, R. Diffusion Models for Robotic Manipulation: A Survey. *Front. Robot. AI* **2025**, *12*, 1606247. [\[CrossRef\]](#) [\[PubMed\]](#)
95. Tricco, A.C.; Lillie, E.; Zarin, W.; O'Brien, K.K.; Colquhoun, H.; Levac, D.; Moher, D.; Peters, M.D.J.; Horsley, T.; Weeks, L.; et al. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Ann. Intern. Med.* **2018**, *169*, 467–473. [\[CrossRef\]](#) [\[PubMed\]](#)
96. Rethlefsen, M.L.; Kirtley, S.; Waffenschmidt, S.; Ayala, A.P.; Moher, D.; Page, M.J.; Koffel, J.B.; the PRISMA-S Group. PRISMA-S: An Extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews. *Syst. Rev.* **2021**, *10*, 39. [\[CrossRef\]](#) [\[PubMed\]](#)
97. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews. *BMJ* **2021**, *372*, n71. [\[CrossRef\]](#) [\[PubMed\]](#)

98. Lee, K.W.; Qin, Y.; Wang, X.; Lim, S.C. DexTouch: Learning to Seek and Manipulate Objects with Tactile Dexterity. *IEEE Robot. Autom. Lett.* **2024**, *9*, 10772–10779. [[CrossRef](#)]
99. Miettinen, K. *Nonlinear Multiobjective Optimization*; International Series in Operations Research and Management Science; Kluwer Academic Publishers: Boston, MA, USA, 1999; Volume 12. [[CrossRef](#)]
100. Li, L.H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.N.; et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 10965–10975. [[CrossRef](#)]
101. Kamath, A.; Singh, M.; LeCun, Y.; Synnaeve, G.; Misra, I.; Carion, N. MDETR-Modulated Detection for End-to-End Multi-Modal Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2021; pp. 1780–1790. [[CrossRef](#)]
102. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 38–55. [[CrossRef](#)]
103. Wu, J.; Antonova, R.; Kan, A.; Lepert, M.; Zeng, A.; Song, S.; Bohg, J.; Rusinkiewicz, S.; Funkhouser, T. TidyBot: Personalized Robot Assistance with Large Language Models. *Auton. Robot.* **2023**, *47*, 1087–1102. [[CrossRef](#)]
104. Das, I.; Dennis, J.E. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems. *Struct. Optim.* **1997**, *14*, 63–69. [[CrossRef](#)]
105. Marler, R.T.; Arora, J.S. The Weighted Sum Method for Multi-Objective Optimization: New Insights. *Struct. Multidiscip. Optim.* **2010**, *41*, 853–862. [[CrossRef](#)]
106. Rockafellar, R.T.; Uryasev, S. Optimization of Conditional Value-at-Risk. *J. Risk* **2000**, *2*, 21–41. [[CrossRef](#)]
107. Ross, S.; Gordon, G.; Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*; JMLR Workshop and Conference Proceedings; PMLR: Norfolk, MA, USA, 2011; pp. 627–635.
108. Mandlekar, A.; Xu, D.; Wong, J.; Nasiriany, S.; Wang, C.; Kulkarni, R.; Li, F.F.; Savarese, S.; Zhu, Y.; Martín-Martín, R. What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Proceedings of the 5th Conference on Robot Learning*, London, UK, 8–11 November 2021.
109. Qin, Y.; Wu, Y.H.; Liu, S.; Jiang, H.; Yang, R.; Fu, Y.; Wang, X. DexMV: Imitation Learning for Dexterous Manipulation from Human Videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2022; pp. 570–587. [[CrossRef](#)]
110. Mandikal, P.; Grauman, K. DexVIP: Learning Dexterous Grasping with Human Hand Pose Priors from Video. In *Proceedings of the Conference on Robot Learning (CoRL)*; PMLR: Norfolk, MA, USA, 2022; pp. 651–661.
111. Kelly, M.; Sidrane, C.; Driggs-Campbell, K.; Kochenderfer, M.J. HG-Dagger: Interactive imitation learning with human experts. In *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2019; pp. 8077–8083. [[CrossRef](#)]
112. Hoque, R.; Balakrishna, A.; Novoseller, E.; Wilcox, A.; Brown, D.S.; Goldberg, K. ThriftyDagger: Budget-Aware Novelty and Risk Gating for Interactive Imitation Learning. In *Proceedings of the 5th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2022; Volume 164, pp. 598–608.
113. Bradley, R.A.; Terry, M.E. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* **1952**, *39*, 324–345. [[CrossRef](#)]
114. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR: Norfolk, MA, USA, 2021; pp. 8748–8763.
115. Shridhar, M.; Manuelli, L.; Fox, D. CLIPort: What and Where Pathways for Robotic Manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*; PMLR: Norfolk, MA, USA, 2022; pp. 894–906.
116. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*; PMLR: Norfolk, MA, USA, 2023; pp. 785–799.
117. Huang, W.; Wang, C.; Zhang, R.; Li, Y.; Wu, J.; Li, F.F. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. In *Proceedings of the 7th Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 229, pp. 540–562.
118. Minderer, M.; Gritsenko, A.; Stone, A.; Neumann, M.; Weissenborn, D.; Dosovitskiy, A.; Mahendran, A.; Arnab, A.; Dehghani, M.; Shen, Z.; et al. Simple open-vocabulary object detection. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2022; pp. 728–755. [[CrossRef](#)]
119. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: New York, NY, USA, 2023; pp. 4015–4026. [[CrossRef](#)]

120. Jiang, Y.; Gupta, A.; Zhang, Z.; Wang, G.; Dou, Y.; Chen, Y.; Li, F.F.; Anandkumar, A.; Zhu, Y.; Fan, L. VIMA: Robot Manipulation with Multimodal Prompts. In *Proceedings of the 40th International Conference on Machine Learning*; PMLR: Norfolk, MA, USA, 2023; Volume 202, pp. 14975–15022.
121. Chao, Y.W.; Yang, W.; Xiang, Y.; Molchanov, P.; Handa, A.; Tremblay, J.; Narang, Y.S.; Van Wyk, K.; Iqbal, U.; Birchfield, S.; et al. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2021; pp. 9044–9053. [[CrossRef](#)]
122. Hampali, S.; Sarkar, S.D.; Lepetit, V. HO-3D_v3: Improving the accuracy of hand-object annotations of the HO-3D dataset. *arXiv* **2021**, arXiv:2107.00887. [[CrossRef](#)]
123. Yang, L.; Li, K.; Zhan, X.; Wu, F.; Xu, A.; Liu, L.; Lu, C. OakInk: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2022; pp. 20953–20962. [[CrossRef](#)]
124. Jian, J.; Liu, X.; Li, M.; Hu, R.; Liu, J. AffordPose: A Large-Scale Dataset of Hand-Object Interactions with Affordance-Driven Hand Pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 14713–14724. [[CrossRef](#)]
125. Ye, J.; Wang, J.; Huang, B.; Qin, Y.; Wang, X. Learning Continuous Grasping Function with a Dexterous Hand from Human Demonstrations. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2882–2889. [[CrossRef](#)]
126. Warnell, G.; Waytowich, N.; Lawhern, V.; Stone, P. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 1545–1553. [[CrossRef](#)]
127. Spencer, J.; Choudhury, S.; Barnes, M.; Schmittle, M.; Chiang, M.; Ramadge, P.; Srinivasa, S. Learning from Interventions: Human–Robot Interaction as Both Explicit and Implicit Feedback. In *Proceedings of the Robotics: Science and Systems XVI, Virtual*, 12–16 July 2020. [[CrossRef](#)]
128. NVIDIA; Mittal, M.; Roth, P.; Tigue, J.; Richard, A.; Zhang, O.; Du, P.; Serrano-Muñoz, A.; Yao, X.; Zurbrugg, R.; et al. Isaac Lab: A GPU-Accelerated Simulation Framework for Multi-Modal Robot Learning. *arXiv* **2025**, arXiv:2511.04831. [[CrossRef](#)]
129. Kim, S.; Cho, J.; Lee, K.W.; Lim, S.C. Pixel2Catch: Multi-Agent Sim-to-Real Transfer for Agile Manipulation with a Single RGB Camera. *IEEE Robot. Autom. Lett.* **2026**, *11*, 9287–9294. [[CrossRef](#)]
130. Zhang, X.; Wang, C.; Sun, L.; Wu, Z.; Zhu, X.; Tomizuka, M. Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning. In *Proceedings of the Conference on Robot Learning*; PMLR: Norfolk, MA, USA, 2023; pp. 1621–1639.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.