Instruct, Don't Rush to Fine-Tune! A Survey on the Art of *Instructing* Your AI for Desired Behaviors

Anonymous authors
Paper under double-blind review

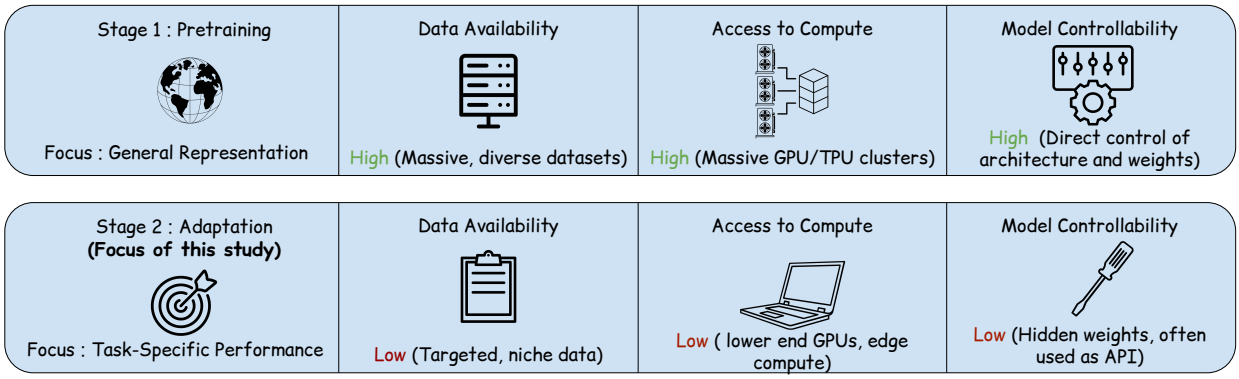
Abstract

Despite the development of large-scale machine learning (ML) models with impressive capabilities, their widespread adoption remains limited by the difficulty in using them out of the box. How to effectively adapt these general-purpose models to application- or user-desired goals and behaviors has emerged as a central challenge. Achieving such reliable and domain-aligned performance often requires deliberate customization beyond simple prompt engineering or naive, expensive fine-tuning. We argue that "smart instruction" plays an important role in enabling efficient and effective adaptation. In this *survey*, we discuss emerging approaches for instructing large scale ML models such as multimodal large language models (LLMs), focusing on how choices in instructional structure affect robustness, generalization, and data efficiency. We introduce *a taxonomy* that characterizes these approaches as distinct modes of instruction and provide a synthesis of common principles, tradeoffs, and design patterns that emerge across domains, from large language models to agentic AI systems and physical AI. We organize this landscape into six complementary instructional forms: *structural, in-context, knowledge-guided, spatial, temporal, and global*, each capturing a distinct way of steering model behavior through reasoning structure, examples, evidence, geometry, procedural sequencing, or long-range context. We conclude with two case studies demonstrating how even practitioners without deep ML expertise can leverage this taxonomy to adapt modern ML systems in real-world settings.

1 Introduction

The rapid advancement of machine learning (ML) over the past decade has been driven largely by three main factors: better architectures, bigger models, and more data. Architectural innovations, while still active, now deliver incremental improvements rather than transformative leaps. While scaling laws show that bigger models and massive training data leads to consistent performance improvements (Kaplan et al., 2020; Hoffmann et al., 2022; Bahri et al., 2024), this approach is increasingly constrained by compute and data costs, environmental impact, and diminishing returns (Villalobos et al., 2024; Wu et al., 2025; Luccioni et al., 2024). Likewise, in terms of more data, the web-scale corpora that underpin today's large models approximate the breadth of collective human knowledge, leaving little room for further gains through raw data accumulation alone. In fields like robotics, where there is room for more data, collecting data is expensive.

These trends reveal a growing consensus: the next frontier in ML progress lies in scaling better with existing data. The saying "garbage in, garbage out" underscores a simple but powerful truth: model capabilities and behaviors are bounded by the quality of their training data and instructions. High-quality, diverse, and task-aligned data is increasingly recognized as primary determinants of generalization, robustness, and alignment. Unlike compute budgets or proprietary architectures, modes of instruction remain the most accessible and controllable levers for practitioners developing novel or domain-specific models.



(a) Two-stage paradigm for large-scale model development and adaptation.

(b) Instruction-centric adaptation taxonomy. The six instruction forms: structural, in-context, knowledge-guided, spatial, temporal, and global, represent complementary ways of steering model behavior during adaptation without retraining from scratch.

Figure 1: Overview of the adaptation setting and the instruction-centric taxonomy used in this survey. (a) Large-scale model development follows a two-stage paradigm: pre-training produces a general-purpose representation under abundant data, compute, and model controllability, whereas adaptation must often operate with limited task-specific data, constrained compute, and low access to model weights. (b) Within this constrained adaptation regime, instruction-centric methods provide practical control mechanisms through structural, in-context, knowledge-guided, spatial, temporal, and global forms of instruction.

Definition (Instructions). Instructions are contextual control variables that bias a model's internal inference trajectory—steering attention, reasoning pathways, and output distributions toward task-aligned behaviors—thereby nudging the model toward a desired style, objective, or interpretation without updating its underlying parameters.

In multimodal large models (MMLMs), such instructions may be expressed through any modality, including text, images, or action signals. By systematically examining how researchers across domains refine, augment, and optimize instructions, we identify shared strategies that improve instruction quality and, consequently, model performance. We especially find and present an instruction-centric taxonomy spanning structural, in-context, knowledge-guided, spatial, temporal, and global forms. Considering these strategies as *instructional levers*, we analyze their empirical trade-offs in quality, cost, and risk. This methodological analysis, therefore, is intended for use as a practical decision checklist that guides how to instruct large-scale ML models to adapt them to specific domains.

2 Background

A growing body of work has examined how to align weakly trained models to general human preferences, for example through preference optimization techniques such as reinforcement learning with human feedback (RLHF) and direct preference optimization (DPO) (Rafailov et al., 2023). More recently, reinforcement fine-tuning (RFT) has extended this paradigm by replacing human preference labels with verifiable reward signals, enabling targeted behavioral updates from correctness feedback alone (Sun et al., 2025). Beyond this initial alignment stage, a wide range of methods has emerged to adapt these human-aligned models to specific applications with minimal additional cost and effort (Li & Liang, 2021; Lester et al., 2021). However, these adaptation strategies and heuristics remain fragmented across the literature and practice. They are often siloed within particular research communities or modalities, leading to inconsistent terminology, overlapping

objectives, and incompatible evaluation protocols. This survey aims to unify this fragmented landscape by introducing a practical, instruction-centric taxonomy for model adaptation.

We bridge the gap between naive prompt engineering (Schulhoff et al., 2024) and fine-tuning (Han et al., 2024), targeting the middle ground of adaptation: methods that make models smarter by shaping the queries to the model, often with minimal or no parameter updates. Our framing unifies techniques currently scattered across natural language processing, vision, speech, and robotics—spanning filtering and selection, labeling and preference design, augmentation and synthesis, and retrieval curation.

Survey methodology: To construct this survey, we reviewed approximately 100 representative works published between 1996 and 2025 spanning classical data-centric learning, large language models, multimodal learning, robotics, retrieval systems, and agentic AI. Historical works are included primarily to provide context and trace the intellectual lineage of modern adaptation techniques, while the survey itself emphasizes recent advances from the foundation-model era. We selected papers that exemplify mechanisms for steering model behavior through instructions, contextual information, external knowledge, geometric structure, temporal organization, or global context, with particular emphasis on influential and representative methods from the last five years. Appendix Table 5 provides a categorized summary of the representative papers and concepts covered in this survey.

3 Classical Perspective: Better Data for Desired Behavior

Prior to the emergence of large-scale models, the statistics and ML community had already established a comprehensive set of data-centric methodologies designed to align model behavior with specific domains, tasks, and data distributions. These methodologies emphasized principled approaches to dataset selection, curation, augmentation, and validation, aiming to ensure consistency between training inputs and deployment environments. As we consider strategies for adapting contemporary models, revisiting these earlier data-centric paradigms is valuable because they provide enduring conceptual and methodological foundations that inform current efforts in instruction tuning, fine-tuning, and domain adaptation of large pretrained systems.

Today, these paradigms remain highly prevalent, particularly during the mid- and post-training stages of current LLMs (Minaee et al., 2024). In these stages, practitioners typically prioritize the strategic selection or collection of high-quality, task-aligned data as the first step to ensure signal over noise. Once a core dataset is established, augmentation techniques are then applied if needed to expand quantity, enhance robustness, or mitigate the costs of further direct data collection.

3.1 Data Augmentation

Scarcity of labeled data was one of the major bottlenecks for earlier ML architectures, particularly before self-supervised learning (Ericsson et al., 2022; Gui et al., 2024). High quality labeled data has been hard to obtain, due to human annotation effort, experimental cost, or safety considerations. In other cases, available data may exhibit distributional biases or insufficient coverage of the target domain (Ye et al., 2024). Thus data augmentation has emerged as a key tool for expanding both the scale and diversity of training corpora without direct data collection. At a high level, augmentation serves three interrelated purposes:

1. Quantity expansion: to compensate for when further data collection is infeasible.

2. Cost mitigation: even when it is possible to collect more data, data collection is constrained by safety and budget (e.g., financial, physical, or time).

3. Robustness enhancement: to expose the model to systematic perturbations that improve invariance to noise, domain shifts, and outliers.

These purposes can be achieved through perturbation or generation based methods.

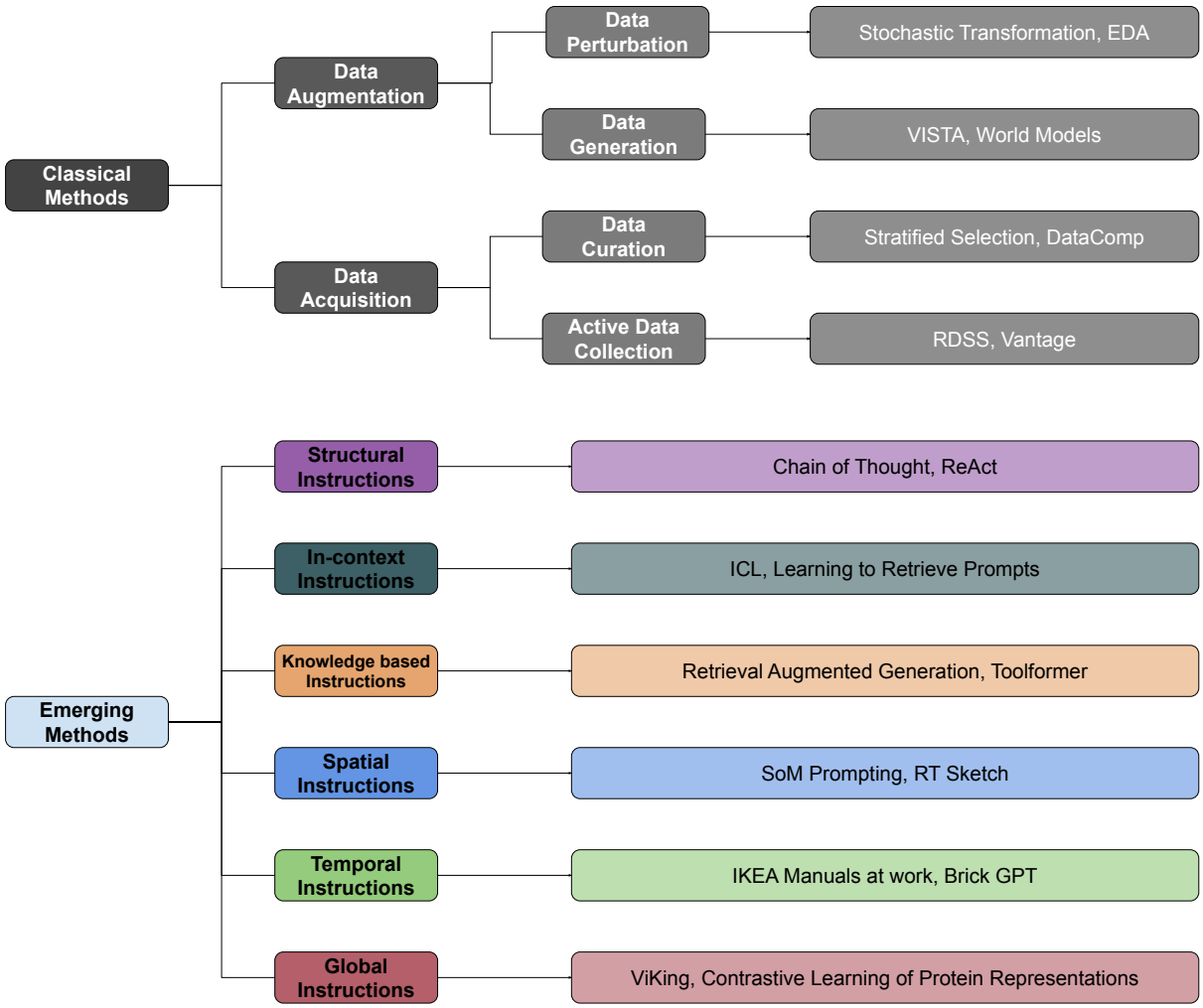


Figure 2: A unified taxonomy of model adaptation strategies spanning classical and instruction-based paradigms. Classical methods (top) divide into two families: data augmentation (perturbation-based and generation-based) and data acquisition (curation and active collection). Emerging instruction-based methods (bottom) comprise six qualitatively distinct modes—Structural, In-Context, Knowledge-Guided, Spatial, Temporal, and Global—each representing an independent axis along which model behavior can be steered at inference time without parameter updates.

3.1.1 Perturbation-based Data Augmentation

The central idea in perturbation is to generate pseudo-novel samples by applying label-preserving transformations to existing samples, thereby approximating the variances expected in the target domain. Conventional augmentation multiplies examples via stochastic transformations (LeCun et al., 1998; Yang et al., 2022; Cubuk et al., 2019). This includes affine transformations, occlusions, or noise injection in visual data; paraphrasing and synonym substitution in textual data; or spectral modifications in audio domains. These methods assume that invariances under specific transformations preserve the semantic class label, thereby improving generalization across minor distributional shifts (Dao et al., 2019).

In computer vision, such transformations included random crops, rotations, translations, mirroring, scaling, or color jittering, simple yet effective perturbations that expanded the apparent diversity of image datasets (Krizhevsky et al., 2012; Takahashi et al., 2019; Howard, 2013). Similarly, in natural language processing, token-level substitutions (Zhang et al., 2015; Wei & Zou, 2019), back-translation (Sennrich et al., 2016),

and paraphrasing methods were adopted to induce lexical and syntactic variation while retaining semantic meaning (Feng et al., 2021). These techniques effectively multiplied data quantity by exploiting low-level symmetries and domain-specific invariances, improving robustness to noise and minor distributional drift.

3.1.2 Generation-based Data Augmentation

Simulation-based augmentation provided an alternative path to data generation grounded in first principles. For instance, simulators backed by physics engines allowed the creation of fully annotated datasets for tasks such as object detection, autonomous navigation, and reinforcement learning (Mayer et al., 2016; James et al., 2019; Shah et al., 2017). These synthetic environments offered controllable diversity to systematically explore distributions beyond what could be collected empirically.

The arrival of generative models such as diffusion (Ho et al., 2020; Rombach et al., 2022) has created unprecedented opportunities for data generation. Such models often enable the efficient synthesis of complex or scarce data at reduced cost. For instance, VISTA (Ren et al., 2025) composes existing video–caption pairs in space and time to create long-duration, high-resolution instruction-following data (VISTA-400K), yielding consistent gains for video models on long-video and video benchmarks after fine-tuning. In robotics, RoVi-Aug (Chen et al., 2024) uses diffusion-based image-to-image generation to rewrite demonstrations, swapping robot embodiments and viewpoints while preserving task semantics, enabling policies to train on diverse embodiments and camera angles and then zero-shot transfer to unseen robots and views. Complementing these, View-Invariant Policy Learning via Zero-Shot Novel View Synthesis (Tian et al., 2024) augments single-view demonstrations with synthesized alternate camera viewpoints using 3D-aware novel-view models, improving robustness to out of distribution cameras without task-specific retraining of the synthesizer. More recently, world models have emerged as a promising direction for generation-based data augmentation, especially in embodied AI, robotics, and autonomous navigation (Hu et al., 2023; Hafner et al., 2023). By learning predictive models of environment dynamics, these systems can synthesize diverse trajectories, counterfactual rollouts, rare-event scenarios, and task-relevant interaction data, reducing the need for costly or risky real-world collection while improving robustness to long-horizon distribution shifts.

3.2 Strategic Data Acquisition

While larger datasets typically improve performance, the indiscriminate inclusion of low-quality or redundant data can dilute signal, slow down optimization, and significantly inflate compute costs. Modern adaptation workflows such as mid-training in LLMs (Gururangan et al., 2020; Parmar et al., 2022), highlight the importance of being cognizant about data quality. By prioritizing examples that maximize novelty or represent under-covered subdomains as well as deciding what not to show a model, we can optimize the "teaching value" of their datasets, ensuring that every training instance provides the greatest possible marginal update to the model's capabilities.

3.2.1 Data Curation

Beyond simple inefficiency, uncontrolled aggregation can also amplify spurious correlations, dataset or source-specific artifacts that models may overfit to instead of task-relevant features (Sorscher et al., 2022; Ye et al., 2024). The result is a paradox of scale: as datasets grow, performance gains diminish while optimization instability, variance, and distributional bias increase (Sorscher et al., 2022). Studies demonstrate that more data do not always equate to better learning (Sorscher et al., 2022), especially when data quality, coverage, or relevance is unevenly distributed.

Consequently, selective data curation, deciding what not to show a model is an equally critical counterpart to augmentation. Active and influence-based selection (Coleman et al., 2019) prioritize instances that maximize novelty or represent under-covered subdomains. Data filtering and quality scoring frameworks such as DataComp (Gadre et al., 2023) and Dolma (Soldaini et al., 2024) formalize this principle at scale.

Data curation optimizes what goes into training rather than indiscriminately scaling: stratify the pool along axes that matter at deployment (task/type, domain/source, difficulty, length, safety), score examples for teaching value (quality, diversity, novelty relative to the current model), and then allocate a fixed budget

across strata to preserve coverage where it matters while suppressing redundancy and spurious shortcuts (Ye et al., 2024).

3.2.2 Active Data Collection

Active data collection addresses the bottleneck of high-quality data by transforming the model from a passive recipient of information into an active participant in its own learning. Instead of collecting data randomly, this approach focuses on acquiring samples that are specifically lacking in the current training distribution or those that provide the most gain when used for training (Coleman et al., 2019). Methods such as RDSS (Shao et al., 2024) and Vantage (Vasudevan et al., 2025) show that collecting a few informative data points leads to better model performance over simply collecting a lot of data points.

4 Modern Perspective: Better Instructions for Desired Behavior

Modern large-scale models possess vast representational capacity and encode substantial amounts of general-purpose knowledge (Hu et al., 2021; Kaplan et al., 2020). However, classical adaptation techniques discussed in Section 3 are often impractical for end users, as they typically lack control over large training datasets or the resources required for expensive retraining. Our focus, therefore, is not on building larger models, but effectively leveraging the capabilities already embedded within them for downstream adaptation. We argue that this can be achieved through instructions introduced in Definition 1. We categorize these instructional paradigms into six broad forms: Structural, In-Context, Knowledge-guided, Spatial, Temporal, and Global, illustrated in Figure 2. This six instruction types capture orthogonal, complementary ways to steer model behavior without parameter updates, covering reasoning, exemplars, external knowledge, geometry, procedural sequencing, and long-range context.

4.1 Structural Instructions

To induce reasoning or efficiently guide outputs at test time, one can *introduce explicit structure into either the prompt or the output-generation framework*. Such structure may be implemented in linear (stepwise), non-linear (branching or hierarchical), or verifiable loop-based forms that iteratively refine and validate intermediate results.

4.1.1 Linear Instructions

Linear structural instructions guide the model in a single, stepwise path from input to answer. Prompt engineering (Sahoo et al., 2024) functions as the control surface by constraining format, roles, and decomposition cues, while Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) elicits explicit intermediate steps that the model would otherwise compress. Brief worked examples and schema prompts establish a locally consistent template, encouraging locality and monotonic progress without branching (Hase et al., 2024). This setting preserves low inference cost and keeps control tight: the user specifies the template and the model follows it to generate the output.

4.1.2 Non-Linear Instructions

Non-linear structural instruction admit multiple concurrent hypotheses that can diverge, backtrack, and recombine. Tree-of-Thought (Yao et al., 2023) and Graph-of-Thought (Besta et al., 2024) instantiate this as a search over partial plans: nodes represent sub-states of reasoning, edges encode expansions or rewrites, and selection policies arbitrate which branches to explore or prune. By mapping problems to a discrete search space with revisitable states, these methods capture combinatorial structure (e.g., symbolic proofs, program synthesis, multi-step planning) and support global consistency checks before committing to a final solution. The prompt describes the search protocol; the model executes it, producing a structured trace rather than a single chain.

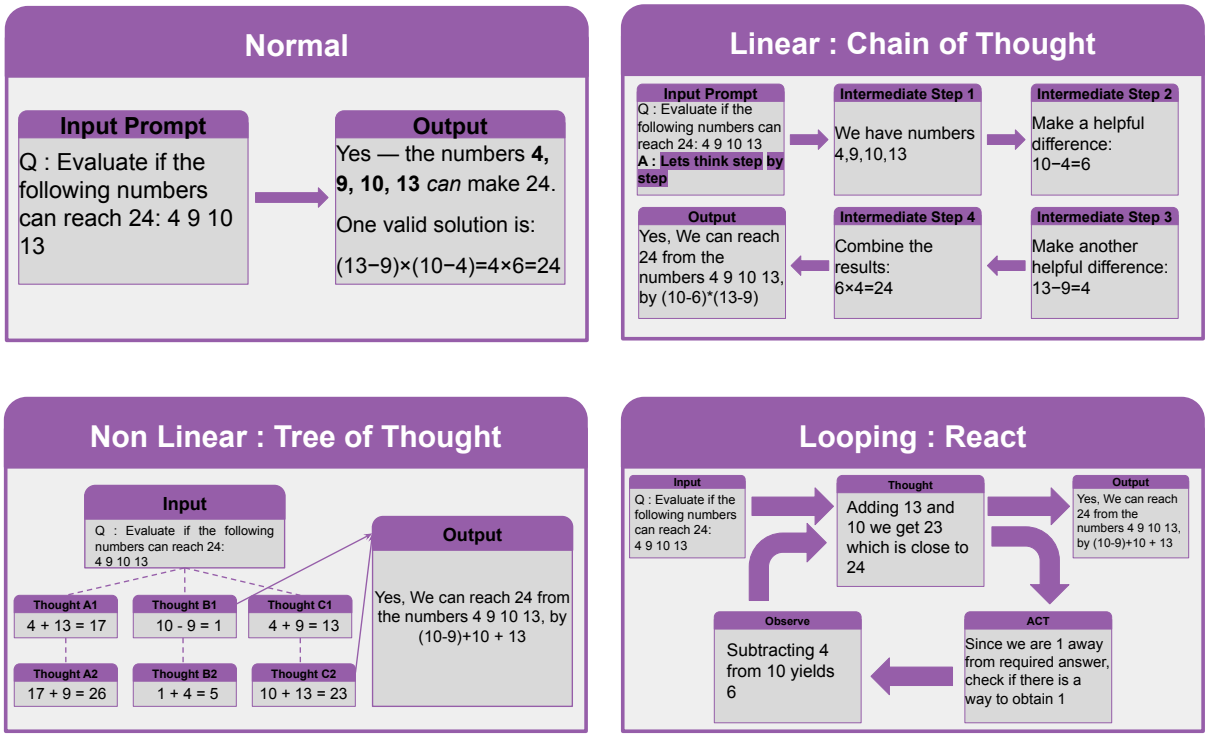


Figure 3: Four structural instruction paradigms applied to the Game of 24 arithmetic task (can 4, 9, 10, 13 produce 24?). (Top-left) A standard prompt yields a direct but opaque answer with no auditable reasoning. (Top-right) Linear Chain-of-Thought (CoT) imposes a sequential decomposition, producing a traceable chain of intermediate arithmetic steps. (Bottom-left) Non-linear Tree-of-Thought branches the search space into parallel hypotheses (e.g., 4+13=17, 10-9=1), pruning dead ends before committing to a final answer. (Bottom-right) Looping ReAct interleaves explicit Thought, Observe, and Output steps in a feedback cycle, enabling self-correction mid-computation. Together, these paradigms illustrate a progression from monolithic to structured, verifiable reasoning, with each successive paradigm offering greater expressiveness at the cost of increased inference overhead.

4.1.3 Looping Instructions

Looping formulations close the reasoning loop with self-evaluation, tools, and environment feedback. Self-consistency (Wang et al., 2022) samples diverse chains and aggregates them to stabilize brittle trajectories; ReAct/Think-Act (Yao et al., 2022) interleave "thoughts" with actions and observations, grounding language decisions in external state; Reflexion (Shinn et al., 2024), Iterative prompt optimization (Karn, 2025), and Self-Refine (Shinn et al., 2024; Madaan et al., 2023) add explicit critique-and-revision cycles. Here, the instruction specifies the controller, the observables, and the stopping rule, turning the model into an iterative solver whose behavior adapts as it inspects, tests, and corrects its own outputs.

4.2 In-Context Instructions

A reliable way to guide a model is often to *provide the task's context, facts, constraints, rubrics, schema, or worked examples, directly within the prompt* to close domain gaps. This approach, known as In-Context Instruction, involves curating a high-information "micro-dataset" that shapes model behavior at inference time without requiring parameter updates. While In-Context Learning (ICL) (Brown et al., 2020) started with simple exemplars to demonstrate decision criteria, contemporary methods incorporate explanations, rationales, or even programs to scaffold the model's internal computations. In-Context instruction relies on a compact, hand-selected set of cases specifically tailored to the current query and domain.

Designing effective in-context instructions entails choices about,

1. *What context to include?* - Information can range from simple label-only examples to comprehensive schemas, rubrics, and graded rationales. The specific ordering and formatting of this content significantly impact both accuracy and calibration (Liu et al., 2023b; Min et al., 2022).

2. *How to select context ?* - Examples can be chosen manually or through automated methods such as embedding-based retrieval (identifying nearest neighbors) (Xu et al., 2023) or active selection (identifying examples that maximize improvement or reduce uncertainty) (Rubin et al., 2022; Zhang et al., 2022). In specialized domains, synthetic demonstrations generated by a teacher model can further expand coverage while maintaining quality through self-critique (Wang et al., 2023b).

3. *How to format context?* - Effective formatting bridges the gap between raw information and executable logic by defining the structural interface the model must navigate (Zhou et al., 2022). Formatting context could take multiple forms:

   - Rubric- or schema-driven prompts specify precise slots and evaluation criteria to ensure consistency across outputs. (Hashemi et al., 2024)
   - Program-of-thought and tool-augmented exemplars demonstrate how to call external tools or follow rigorous procedural steps (Chen et al., 2023). Note that while this is similar to structural instruction, the structure provided is dynamic, allowing the model to adapt its logic or tool usage based on the problem at hand.
   - Constitution-style rubrics encode safety and preference constraints that the model applies to its own outputs through self-critique (Bai et al., 2022; Gou et al., 2024).

These strategies augment the prompt with a small number of carefully chosen cases that steer the model toward the constraints and invariances that matter at deployment.

For example, consider the task of classifying a patient message into Emergency, Urgent, Routine. The prompt includes: (i) what context — an explicit rubric, worked exemplars, rationales, and a fixed output schema; (ii) how it is selected — examples spanning decision boundaries with balanced severity and diversified symptoms (optionally retrieved via embedding similarity); and (iii) how it is formatted — rubric first, consistent Patient → Label → Rationale structure, and explicit output specification. With ICL, the model aligns to rubric-defined decision criteria and produces calibrated, structured outputs. Without ICL, predictions rely on generic priors, often leading to inconsistent reasoning and weaker calibration. Beyond classification tasks, ICL has been applied in assistive robotics to personalize LLM behavior for cognitively diverse users, enabling stage-aware interactions without domain-specific retraining (Smaili et al., 2026).

4.3 Knowledge-Guided Instruction

Knowledge-guided instructions *direct the model to read from additional sources of knowledge before or during output generation* to close knowledge gaps and prevent hallucinations. While in-context instructions align the model input with domain, knowledge-guided instructions provide explicit information required to generate a truthful output. By specifying which sources to consult, how to stage them, and when to trust them, generation is conditioned on explicit evidence rather than parametric recall. Classical RAG (Lewis et al., 2020) is the canonical instantiation of this principle. Contemporary efforts in developing episodic and long-term memories (Wang et al., 2023a; Fountas et al., 2025; Ji-An et al., 2024) further let the model read its own past conversations, tool outputs and intermediate computations before answering. As another method, kNN-style caches (Safaya & Yuret, 2024) retrieve neighboring states, injecting "experience" directly into decoding.

Beyond static corpora, tool-mediated instructions allow models to actively query external environments to fetch "just-in-time" data (Schick et al., 2023; Nakano et al., 2022). By integrating APIs, calculators, and web search into the instruction loop, the model's context is no longer limited to the training cutoff or a local vector store. Instead, tool calling acts as a specialized data acquisition layer where the model must learn to transform a user's high-level intent into precise queries, then filter and synthesize the resulting "fresh"

Example: Regular vs. In-Context Instruction for Clinical Triage Classification

Regular prompt

```
Classify "I have sharp abdominal pain and can't keep food down" into Emergency,
Urgent, or Routine.
```

ICL prompt

```
You are a clinical triage assistant.
Urgency Rubric:

    - Emergency:  Immediate life-threatening risk (e.g., chest pain, breathing
      difficulty).

    - Urgent:  Requires same-day medical attention but not immediately
      life-threatening.

    - Routine:  Can wait for scheduled appointment.

Output format:
Label:  <Emergency | Urgent | Routine>
Rationale:  <1-2 sentence justification referencing rubric>

Examples:
Patient:  "I have crushing chest pain and feel dizzy."
Label:  Emergency
Rationale:  Chest pain with dizziness indicates possible cardiac emergency.

Patient:  "My fever has been 101°F for two days."
Label:  Urgent
Rationale:  Persistent fever suggests infection requiring timely evaluation.

Patient:  "I need a refill of my allergy medication."
Label:  Routine
Rationale:  Medication refill is non-acute and can be handled routinely.

Now classify:
Patient:  "I have sharp abdominal pain and can't keep food down."
```

data into a grounded response. This shifts the adaptation challenge from storing knowledge to mastering the interfacing logic: knowing which external data source is authoritative for a given task and how to resolve contradictions between tool outputs and internal parametric knowledge.

Agent runtimes such as Clawdbot (OpenClaw) (OpenClaw, 2026) illustrate an applied instantiation of knowledge-guided instruction in interactive settings. Rather than limiting retrieval to static corpora or vector databases, these systems expose the model to structured environmental state, tool interfaces, and persistent memory through a controlled execution layer. The language model operates as a planner that utilizes that knowledge obtained by issuing tool calls and receiving observations. Here the knowledge is not just external documents but also includes executable affordances and intermediate state transitions.

4.4 Spatial Instructions

As the saying goes, "a picture is worth a thousand words," spatial instructions *treat geometry as a language of intention*. Considering the challenges in describing visual information in natural language, we can transform human spatial intuition into machine-interpretable signals by querying with visual markers such as arrows,

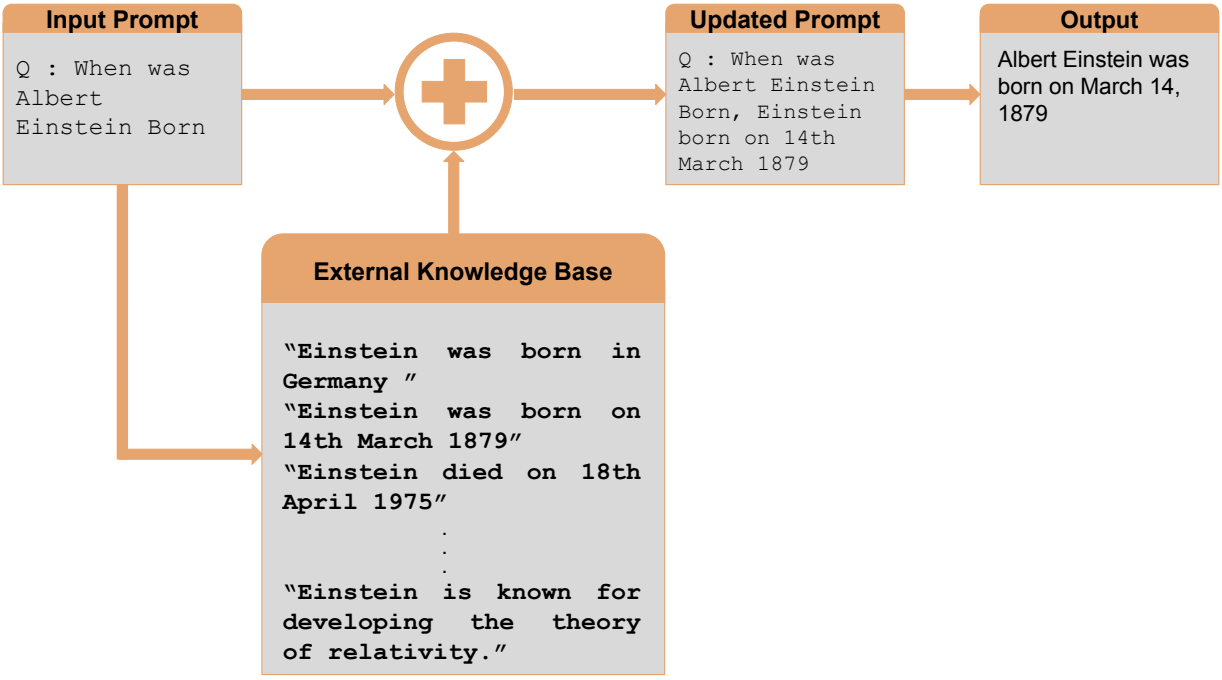


Figure 4: Schematic of Retrieval-Augmented Generation (RAG) as a canonical knowledge-guided instruction mechanism. Given an input prompt containing a factual query ("When was Albert Einstein born?"), semantically relevant passages are retrieved from an external knowledge base and concatenated with the original prompt to form an updated, evidence-enriched context.

contours, goal regions. **Semantic and instance segmentation** layers (Long et al., 2015; He et al., 2017; Csurka et al., 2022; Hafiz & Bhat, 2020), overlaid in a different color or by contours, act as instructional masks that tell the model exactly which pixels belong to the regions of interest versus the background (Zou et al., 2023; Acuna et al., 2018). **Depth maps** provide explicit information about the scene's 3D structure. In robotics, augmenting RGB observations with predicted depth (e.g., via RGB-to-Depth models (Chen et al., 2018)) exposes geometric cues such as relative distance, surface orientation, and occlusion. This additional modality helps models reason about reachability, clearance, and proximity, enabling more precise and collision-aware motion planning (Jacquet & Alexis, 2024; Mahler et al., 2017).

Enumerating objects facilitates object-level grounding in natural language instructions. By assigning discrete identifiers to scene elements, models can resolve references unambiguously. For instance, Set-of-Mark (SoM) prompting (Yang et al., 2023) enables the model to refer to indexed objects (e.g., "pick object 2"), reducing ambiguity and improving spatial reasoning.

Adding arrows, waypoints, or trajectories helps to encode information about motion and physical forces acting on different parts of the scene depicted in the image. For example, Robotic Visual Instruction (Li et al., 2025) uses arrows to indicate the direction in which the robot should move to complete the task successfully.

Marking target regions enables visually encoding the physical boundaries of required actions. For instance, Spatial Diagrammatic Instructions (SDI) (Sun et al., 2024) uses a variety of geometric markers drawn over images to indicate motion or targets which provide immediate, low-level control signals. Policies trained on SDI data outperform text-conditioned baselines in fine-grained manipulation and navigation, showing that diagrammatic input naturally conveys hierarchical reasoning: where to move, how to move, and when to stop.

Drawing Sketches and Diagrams provide a compact, expressive channel for specifying where and how the model should take action. By drawing directly onto the visual input, these instructions provide a direct bridge

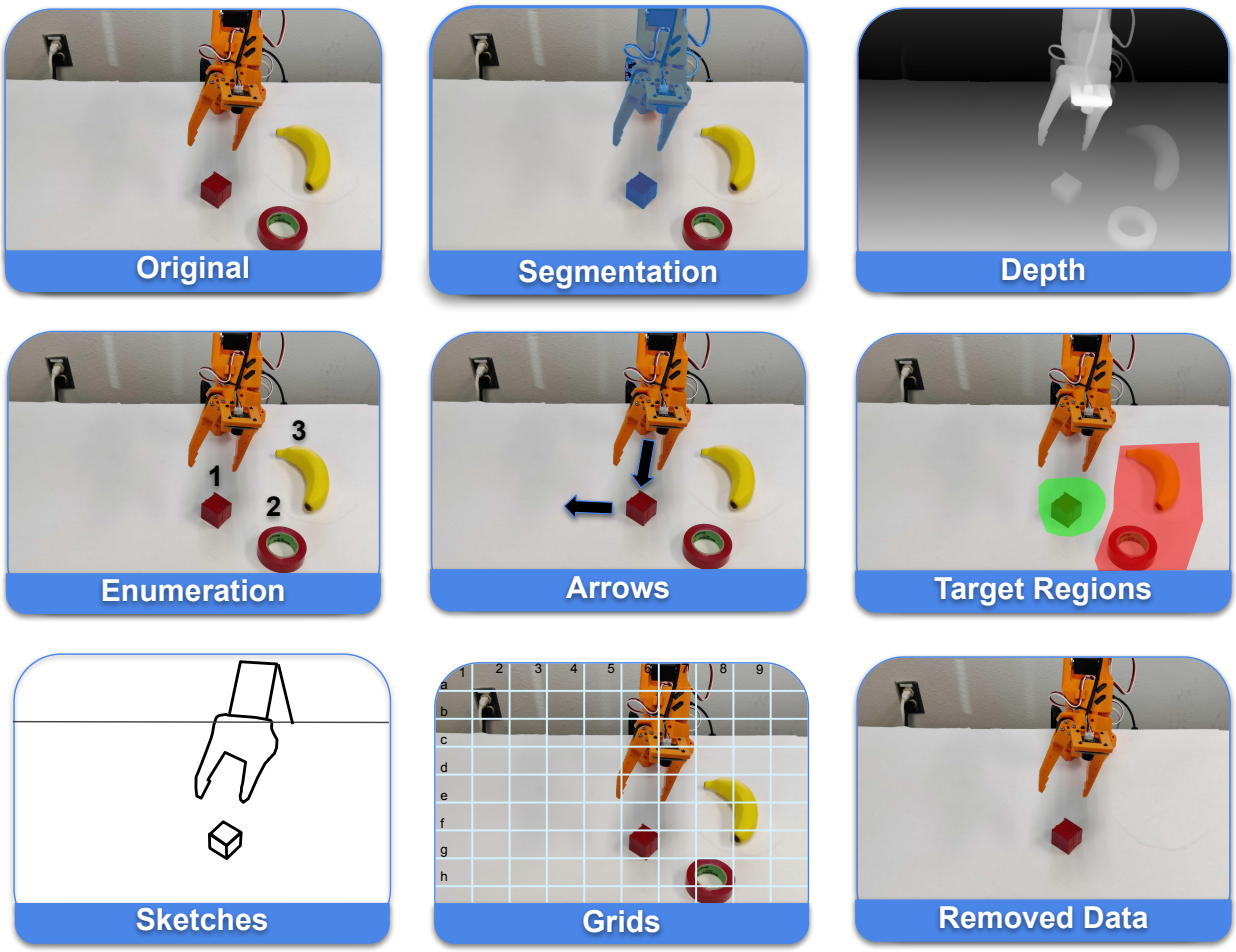


Figure 5: Nine canonical spatial instruction modalities illustrated on a common robotic tabletop manipulation scene. Each panel applies a distinct transformation to the same RGB observation: (1) Original : the unmodified RGB baseline; (2) Segmentation : pixel-level masks isolating task-relevant objects; (3) Depth : a grayscale map encoding scene geometry and occlusion structure; (4) Enumeration : discrete numeric identifiers overlaid on objects for unambiguous referencing (cf. Set-of-Mark prompting); (5) Arrows : directional motion cues communicating intended robot movement; (6) Target Regions : color-coded overlays delineating goal areas; (7) Sketches : freehand diagrammatic drawings providing compact geometric intent; (8) Grids : a coordinate-cell overlay that discretizes the continuous visual space into symbolic positions; and (9) Removed Data : selective masking of task-irrelevant regions to suppress nuisance variance. These modalities collectively illustrate how geometric and semantic intent can be injected directly into a model's visual input stream without modifying its weights.

from human mental models to image-space coordinates. RT-Sketch (Sundaresan et al., 2024) introduces a goal-conditioned imitation framework that augments visual observation with human-provided sketches. By concatenating the drawn goal map with camera images, the modified RT-1 transformer learns to interpret hand-drawn cues as geometric constraints. This enables robust task completion in ambiguous environments where linguistic instructions fail (e.g., "move the block near the edge"). Notably, the sketches act as dense supervision: they encode continuous, spatially grounded intentions, unlike text which is symbolic and sparse.

Grids overlaid on an input image transform a continuous visual scene into a structured, discrete state space. This discretization enables symbolic reasoning over visual inputs, allowing trajectories to be represented as sequences of grid cells (e.g., A1→B1→B2→B3→C4) (Wang et al., 2023c; Huang et al., 2023; Shridhar et al., 2023).

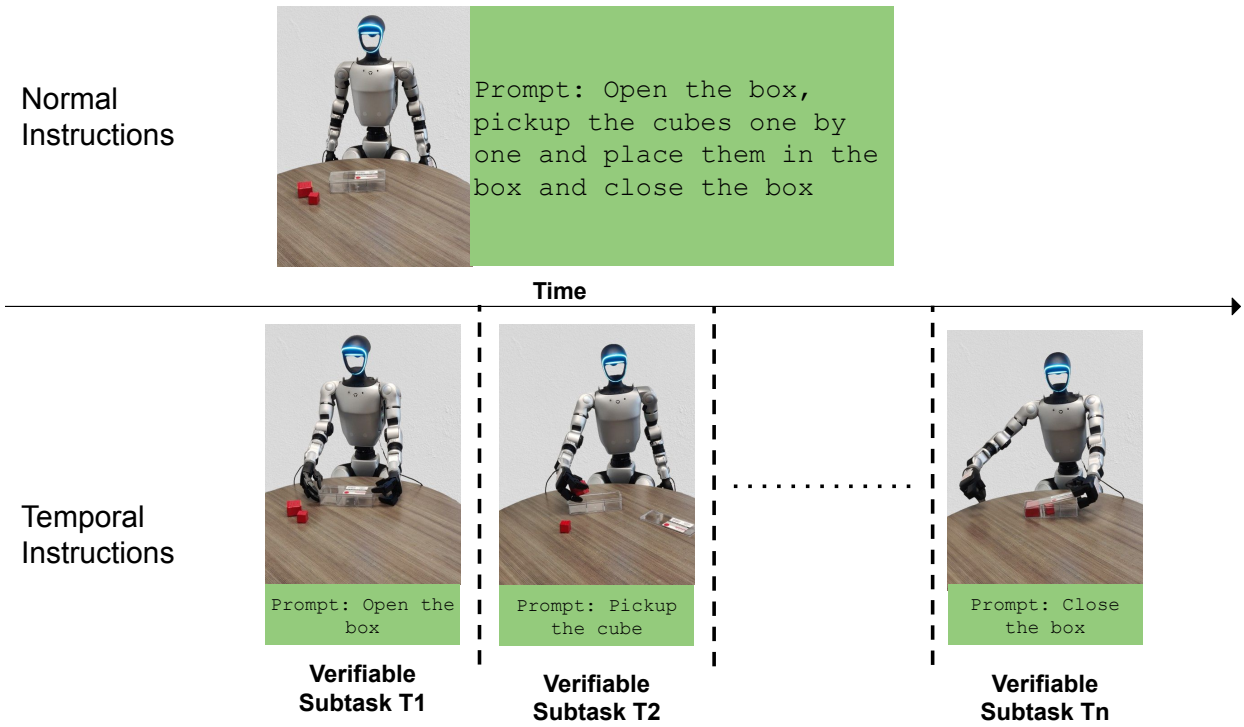


Figure 6: Comparison of monolithic versus temporally-structured instructions on a robotic box-packing task. (Top) A normal instruction delivers the entire task as a single prompt ("Open the box, pick up the cubes one by one, place them in the box, and close the box"), providing the model with no verifiable intermediate structure and no mechanism for partial-task recovery. (Bottom) Temporal instructions decompose the identical task into a sequence of discrete, independently-verifiable subtasks (T1: Open the box; T2: Pick up the cube; ... Tn: Close the box), with explicit checkpoints along the time axis.

Information removal realizes the saying "Less is sometimes more." Removing noise from the data leading to better models is a well studied phenomenon (Tishby et al., 2000; Bengio et al., 2013; Zhou et al., 2023). This also applies to spatial instructions. Instructing a model where not to look can be as informative as telling it what to attend to. By selectively masking, cropping, or lightly inpainting task-irrelevant regions, we reduce nuisance variability and steer predictions toward task-causal evidence, without adding overhead during the data collection phase. BYOVLA (Hancock et al., 2025) instantiates this idea at test time with no weight updates. A VLM proposes distractors, segmentation grounds them to pixel masks, and a visual sensitivity probe perturbs each candidate to measure action deviation; only regions that are irrelevant yet action-sensitive are minimally edited (e.g., object inpainting or background recoloring) before passing the image to the policy. This run-time pipeline is model-agnostic and requires neither fine-tuning nor access to weights.

4.5 Temporal Instructions

Temporal instructions formalize "when to do" along with what to do: sequences of steps, intermediate states, and sequential conditions that make the action structure explicit. When instructions are represented as temporally ordered traces such as assembly manuals, multi-step demonstrations, or action schema, the input becomes an algorithm where each step is both supervision and constraint. This temporal scaffold teaches models to anticipate state transitions, resolve ambiguities, and compose tasks. This also allows the user to have a verification step between each state which leads to overall safer models.

IKEA Manuals at Work (Liu et al., 2024) illustrates that instruction manuals can be converted into sequential teaching signals by aligning textual steps ("attach panel A to side B") with visual states from 3D assets

Figure 7: Ego-centric local observations versus top-down global instructions in vision-language navigation. (Left) Four street-level views from the vehicle's perspective capture rich local context—lane markings, stop signs, building facades—but are inherently limited in spatial scope and cannot encode the intended destination or global route structure. (Right) A map-level view provides global instruction as a planned route (blue path), encoding long-range spatial dependencies, goal context, and topological structure that are unobservable from any individual local frame.

and time-synced video frames. These (instruction step, image) pairs create a temporally ordered training corpus. Training on these traces promotes transition-aware representations: the model learns precursors and consequences of actions.

The Brick-GPT (Pun et al., 2025) paradigm extends this procedural supervision into the symbolic domain. Here, the act of assembling LEGO bricks is represented as a sequence, where each brick could be considered a token and each structure a sentence. The model learns to generate coherent "build sequences" that respect spatial and structural constraints, demonstrating emergent planning behavior. This illustrates that temporal instructions are data with topology. Here each assembly task becomes a miniature algorithm, interpretable, modular, and transferable to other domains.

4.6 Global Instructions

Global context can be defined as information that is not inferable from the immediate scope of the data but observable from higher-level structure, long-range dependencies, or the broader environment in which the data is embedded. Across domains, the inclusion of global context, whether spatial, structural, or semantic, consistently improves model performance.

ViKiNG (Shah & Levine, 2022) extends the scope of robotic navigation from room-scale to kilometer-scale trajectories by integrating global geographic priors into a vision-language navigation model (Kim et al., 2024; Hirose et al., 2025). Unlike traditional navigation systems trained on local viewpoints or turn-by-turn instructions, ViKiNG introduces a form of global instruction by embedding global geographic cues (e.g., aerial maps, street-level orientation, GPS hints) alongside egocentric observations. Empirically, these global hints produce dramatic gains in route planning and localization accuracy. The model learns not merely to see but to reason, linking visual landmarks to their broader spatial topology.

A similar principle could be observed in protein research. Self-supervised protein models trained via mutual information maximization (Ozsoy et al., 2022) exploit global biological context: residue-level embeddings are trained not in isolation but in relation to the global fold of the protein. By maximizing the mutual

information between local amino acid windows and the entire protein representation, the model implicitly learns the structural grammar of biochemistry without explicit labels. This global-local coupling lead to each residue's meaning being derived not from its sequence alone but from its role in the larger structural narrative. Models trained in this manner outperform baseline encoders on downstream tasks such as stability prediction, epitope identification, and mutational robustness, confirming that global coherence acts as a natural supervisory signal.

4.7 Combining Multiple Methods

Compositions of multiple instructional methods, where different types of instructions are layered to constrain behavior across complementary dimensions, often lead to improved model performance. These hybrid designs treat instruction not as a monolithic prompt, but as a coordinated control stack that operates over perception, reasoning, memory, and action. For example, Ego-Zero (Liu et al., 2025) demonstrates how combining egocentric visual observations (spatial instructions) with environmental information (In-Context instructions) allows models to generalize to novel environments without task-specific retraining. Similarly, HAND (HAND Me the Data) (Hong et al., 2025) exemplifies this synergy by combining spatial instructions, using human-centric data to provide dense geometric supervision, with structural instructions like Embodied Chain-of-Thought. These hybrid approaches show that while individual methods like spatial cues disambiguate perception or temporal traces enforce compositionality, their true power lies in how they compensate for each other's limitations to transform large models into adaptive, multi-constraint systems. Across these examples, a recurring pattern emerges: each instructional method compensates for the limitations of others. Structural instructions stabilize reasoning, in-context examples localize behavior, knowledge-guided retrieval anchors generation in evidence, spatial cues disambiguate perception, temporal traces enforce compositionality, and global context aligns local decisions with long-horizon objectives.

Viewed through this lens, instruction becomes a form of system design rather than prompt design. The central challenge is no longer choosing the "best" instruction, but determining how to compose instructional signals across modalities and timescales to match the structure of the task. This perspective motivates future work on principled instruction composition, automated selection of instructional stacks, and evaluation protocols that reflect real-world, multi-constraint deployment settings. More broadly, there is already a rich literature on automatically selecting and composing learning systems, ranging from the classical algorithm selection problem (Rice, 1976) to AutoML (Feurer et al., 2015) systems such as auto-sklearn and more recent compilers for optimizing LM pipelines such as DSPy (Khattab et al., 2023) and these ideas provide a natural foundation for automatically choosing and composing instructional stacks as well.

5 A Practitioner-Oriented Framework for Selecting Instructional Strategies

The successful adaptation of large models relies on the amalgamation of appropriate instructional strategies with high-quality data. While the taxonomy detailed in Section 4 explores the underlying mechanics of these methods, the practical selection of an instructional lever is primarily governed by two factors: the specific failure mode observed in the model and the nature of the data available to the practitioner. Table 1 summarizes this mapping between observed failure modes, available data, and suitable instructional strategies.

When a model fails to meet desired performance thresholds, and the practitioners can categorize the failure into any of the following, they can match it to a targeted instructional strategy.

1. **Reasoning Inconsistency** - If the model exhibits ambiguous reasoning or inconsistent logic but the user only has a high-level problem description, structural instruction is the primary lever. By imposing a reasoning structure, one can improve the consistency and traceability of the output.

2. **Domain Gaps** - When the base model is already capable, as with an LLM, but shows limited adaptation to a particular domain, such as a specific style of exam question, a small set of representative examples or grading rubrics in the prompt (In-Context instructions) can steer it toward domain-specific conventions without requiring parameter updates.

Table 1: Mapping between problem characteristics, available data, and instructional strategies. Instruction acts as an input-space control mechanism that shapes model behavior without retraining. See Appendix Table 4 for *a comparative synthesis*.

Observed Problem	**Data Available**	**Instruction Type**	**Effect of Instruction**
Ambiguous reasoning or inconsistent logic	Problem description only; no intermediate structure	Structural Instruction	Imposes reasoning structure (decomposition, search, or iterative refinement), improving consistency and traceability.
Limited domain adaptation despite capable base model	Small number of representative examples or rubrics	In-Context Instruction	Creates a task-local training distribution that guides behavior without parameter updates.
Hallucination or lack of factual grounding	External documents, databases, or prior outputs	Knowledge-Guided Instruction	Anchors generation in explicit evidence and reduces reliance on parametric recall.
Ambiguous spatial intent or perception mismatch	Images, sketches, masks, or geometric cues	Spatial Instruction	Transforms geometric intent into machine-interpretable constraints, reducing ambiguity.
Multi-step tasks or procedural failures	Step sequences, demonstrations, or manuals	Temporal Instruction	Introduces ordered constraints enabling compositional reasoning and intermediate verification.
Locally correct but globally inconsistent behavior	Long-range context, maps, or global structure	Global Instruction	Aligns local decisions with long-horizon objectives and global coherence.

3. **Hallucination** - In cases of hallucination (Ji et al., 2023; Huang et al., 2025) or lack of factual grounding (Honovich et al., 2021), knowledge-guided instruction should be employed. By providing external documents or databases, the model can be anchored in explicit evidence, significantly reducing its reliance on potentially outdated or incorrect parametric recall.

4. **Misaligned Spatial Intent** - When there is a mismatch between spatial intent and perception, for instance in vision based manipulation tasks in robotics, spatial instruction transforms human geometric intuition, via sketches, masks, or cues, into machine-interpretable constraints.

5. **Flawed procedural suggestions** - For multi-step tasks or procedural failures, using temporal instruction helps introduce ordered constraints. This enables compositional reasoning and provides discrete points for intermediate verification.

6. **Long Range Inconsistency** - When a model's behavior is locally correct but fails to align with long-range dependencies, such as geographic priors, or the broader environmental structure, using global instructions can provide the necessary context required to mitigate this issue.

A comparative synthesis of the six instructional forms, including their core principles, typical use cases, failure modes, costs, robustness profiles, and open challenges, is provided in Appendix Table 4.

6 Case Study Experiments

To evaluate the practical utility of the instructional approaches introduced in this work, we conducted two case studies focused on spatial instructions. Both studies examine whether spatially grounding a model's inputs can meaningfully improve task performance: the first on a visual spatial reasoning benchmark where the model must verify directional relations between objects, and the second on an out-of-distribution generalization benchmark where spurious background cues otherwise mislead image classifiers.

Table 2: Accuracy of VLM-only vs. SoM-augmented spatial verification on the VSR zero-shot test split (Liu et al., 2023a), broken down by directional relation. **VLM**: InstructBLIP-Vicuna-7B (Dai et al., 2023) on the raw image. **VLM + SoM**: YOLOv8 + SAM (Kirillov et al., 2023) numbered overlays with geometric rule-based verification (75.0% accuracy on 64% of samples where both entities are detected; VLM fallback on the remainder). Best result per row in **bold**.

Spatial Relation	Number	VLM Only	VLM + SoM
left of	15	46.7%	**86.7%**
at the left side of	25	40.0%	**64.0%**
above	26	50.0%	**69.2%**
at the right side of	17	58.8%	**64.7%**
beneath	43	48.8%	**55.8%**
below	19	**63.2%**	52.6%
right of	5	40.0%	40.0%
Overall	150	50.0%	**62.7%**

6.1 Case Study 1: Spatial instructions for visual spatial reasoning

To evaluate the benefit of spatial instruction via Set-of-Mark (SoM) prompting (Yang et al., 2023), we conduct an experiment on the Visual Spatial Reasoning (VSR) benchmark (Liu et al., 2023a), which tests whether a model can verify directional spatial-relation statements (e.g., *"The car is to the left of the suitcase"*) from natural image–caption pairs. We evaluate 150 balanced True/False samples from the zero-shot test split, covering seven directional relations (*left of*, *right of*, *above*, *below*, etc.), using InstructBLIP-Vicuna-7B (Dai et al., 2023) as the shared vision–language model.

The VLM-only baseline receives the raw image and answers a yes/no reformulation of the caption; it achieves 50.0% accuracy, equivalent to random chance, reflecting the model's inability to reliably resolve fine-grained directional relations from unstructured pixel input. The SoM-augmented pipeline first runs YOLOv8 and SAM (Kirillov et al., 2023) to generate numbered, color-coded object overlays with explicit 2D position tags (e.g., `[1] car (left-top), [2] suitcase (right-middle)`). When both the subject and object of the caption are detected—64% of samples—the stated relation is verified geometrically from bounding-box positions without any VLM call, achieving 75.0% accuracy. For the remaining 36% where YOLO misses an entity, the annotated image and object manifest are passed to the VLM as a fallback. Overall, SoM grounding raises accuracy to 62.7%, a +12.7 percentage-point gain over the VLM baseline (Table 2). The strongest improvements occur for the clearest 2D positional relations: *left of* (+40.0 pp) and *at the left side of* (+24.0 pp), demonstrating that enumerating and spatially tagging detected objects converts an ambiguous pixel-level reasoning problem into a structured, verifiable computation—precisely the mechanism that spatial instructions are designed to enable.

6.2 Case Study 2: Spatial Instructions for Reducing Out Of Distribution Errors

We evaluate whether spatial instructions can reduce shortcut learning on the *Spawrious* benchmark (Lynch et al., 2023), which tests whether image classifiers rely on spurious background cues instead of the foreground dog breed. We compare ResNet-18 and ResNet-50 (He et al., 2016) trained on either original images or segmented images obtained using SAM masks (Kirillov et al., 2023), where the background is removed and only the dog is retained.

We report results on the many-to-many (m2m) out of distribution (OOD) splits of Spawrious. In this setting, multiple breeds are spuriously correlated with multiple backgrounds during training. The *easy*, *medium*, and *hard* splits correspond to increasing levels of distribution shift and increasing difficulty of breaking these correlations at test time.

As shown in Table 3, training on segmented inputs substantially improves OOD accuracy across all settings. This suggests that removing background information acts as an effective spatial instruction, forcing the model to rely more on breed-relevant foreground features rather than background shortcuts.

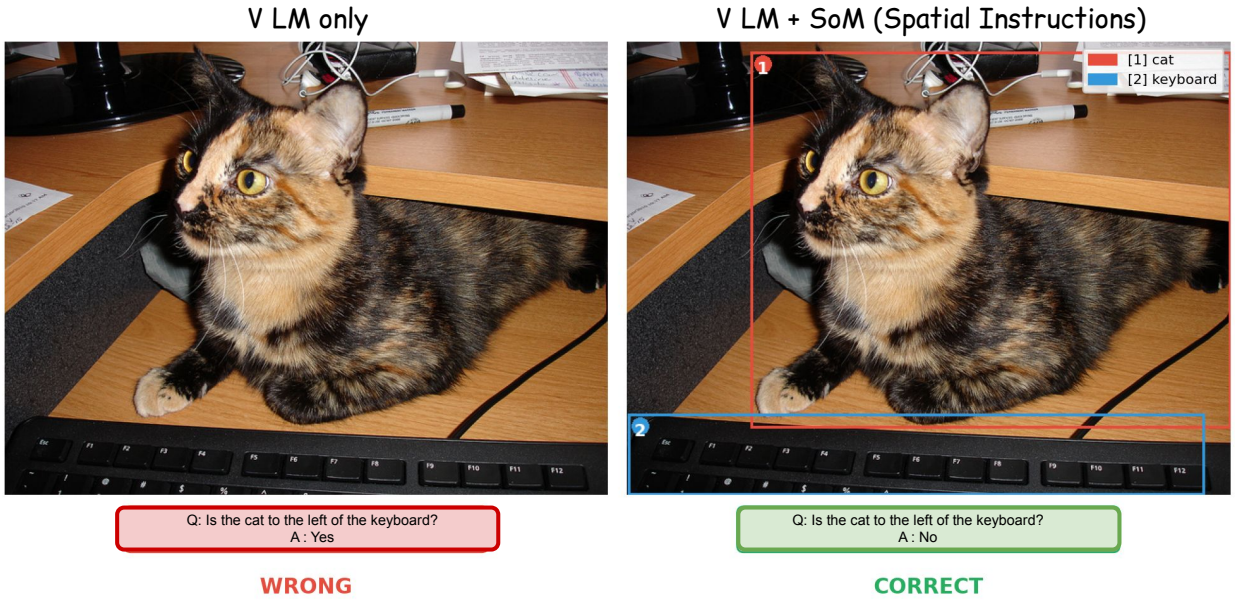


Figure 8: Qualitative example of visual spatial reasoning with Set-of-Mark (SoM) prompting. Left: The VLM-only baseline, given the raw image and the query "Is the cat left of the keyboard?", predicts the relation incorrectly, reflecting the ambiguity of resolving object references and directional relations directly from unstructured pixels. Right: The SoM-augmented pipeline overlays explicit object identifiers and geometric position cues, converting the same query into a grounded spatial verification problem over detected entities (e.g., cat and keyboard).

Table 3: OOD test accuracy on the Spawrious many-to-many (m2m) benchmark under original training and spatially instructed training via foreground segmentation. In the m2m setting, multiple breeds are spuriously correlated with multiple backgrounds during training. The *easy*, *medium*, and *hard* splits correspond to increasing levels of spurious-correlation difficulty and distribution shift at test time. Results are reported as mean ± standard deviation over 3 runs.

	ResNet-18		**ResNet-50**	
Benchmark split	**Orig**	**Spatial Inst.**	**Orig**	**Spatial Inst.**
`m2m-easy`	50.66 ± 2.72	85.10 ± 8.13	45.97 ± 10.61	87.37 ± 11.70
`m2m-medium`	30.55 ± 6.38	80.56 ± 12.92	30.89 ± 6.01	85.12 ± 2.71
`m2m-hard`	27.10 ± 10.12	70.74 ± 4.97	18.08 ± 7.03	82.27 ± 3.53

7 Conclusions

In this survey, we argued that effective instruction is often a practical and scalable lever for adapting large models to domain- and user-specific goals. We unified a fragmented literature by introducing an instruction-centric taxonomy spanning structural, in-context, knowledge-guided, spatial, temporal, and global forms, and by highlighting recurring design patterns and trade-offs that determine robustness, generalization, and data efficiency in real deployments. Building on this taxonomy, we provided a practitioner-oriented framework that maps common failure modes to targeted instructional strategies, emphasizing that instruction should be treated as system design. Future work will explore automated selection and composition of instructional strategies, dynamic multi-modal instruction integration, and standardized community-driven benchmarks to systematically evaluate trade-offs across instruction types.

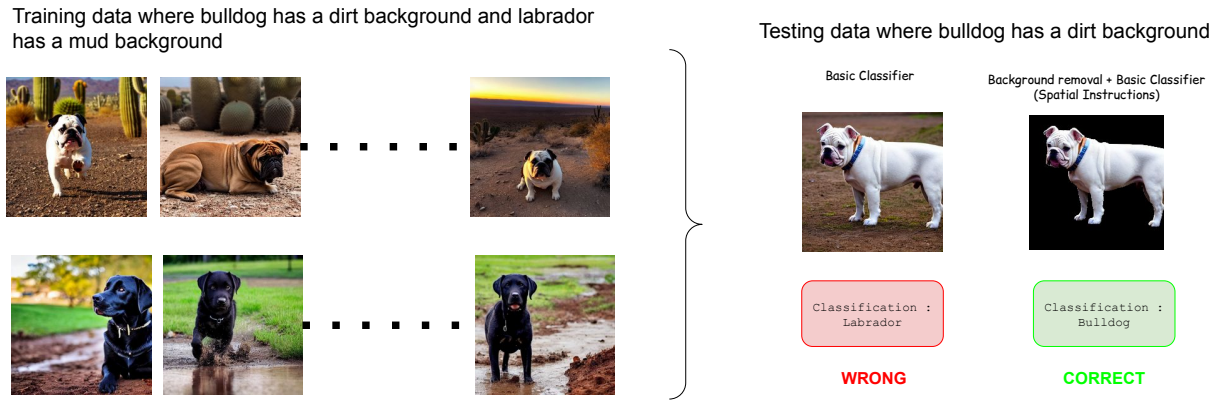


Figure 9: Qualitative example of reducing out-of-distribution errors through spatial instruction via foreground segmentation. Left: In the training distribution, dog breeds are spuriously correlated with background environments, encouraging the classifier to rely on shortcut cues rather than breed-specific foreground features. Right: At test time, this shortcut leads a standard classifier to misclassify a bulldog presented against an unfamiliar background. When the background is removed through segmentation, the resulting spatially instructed input suppresses nuisance context and preserves only the task-relevant foreground object, enabling correct classification.

References

David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 859–868, 2018.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *arXiv preprint arXiv:2409.03403*, 2024.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2023. *URL https://arxiv.org/abs/2211.12588*, 53, 2023.

Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating depth from rgb and sparse sensing. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 167–182, 2018.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. Semantic image segmentation: Two decades of research. *Foundations and Trends in Computer Graphics and Vision*, 14(1-2):1–162, 2022.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International conference on machine learning*, pp. 1528–1537. PMLR, 2019.

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pp. 968–988, 2021.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

Zafeirios Fountas, Martin A Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou Ammar, and Jun Wang. Human-inspired episodic memory for infinite context llms. In *International Conference on Learning Representations*, volume 2025, pp. 77230–77268, 2025.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, volume 2024, pp. 57734–57811, 2024.

Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8342–8360, 2020.

Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Asher J Hancock, Allen Z Ren, and Anirudha Majumdar. Run-time observation interventions make vision-language-action models more visually robust. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9499–9506. IEEE, 2025.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks, 2024. *URL https://arxiv. org/pdf/2401.06751. pdf*, 2024.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13806–13834, 2024.

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

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *arXiv preprint arXiv:2509.19480*, 2025.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Matthew Hong, Anthony Liang, Kevin Kim, Harshitha Rajaprakash, Jesse Thomason, Erdem Bıyık, and Jesse Zhang. Hand me the data: Fast robot adaptation via hand path retrieval. *arXiv preprint arXiv:2505.20455*, 2025.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. Q2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7856–7870, 2021.

Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model complexity of deep learning: A survey. *Knowledge and Information Systems*, 63(10):2585–2619, 2021.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55, 2025.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Martin Jacquet and Kostas Alexis. N-mpc for deep neural network-based collision avoidance exploiting depth images. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13536–13542. IEEE, 2024.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12627–12637, 2019.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Li Ji-An, Corey Y Zhou, Marcus K Benna, and Marcelo G Mattar. Linking in-context learning in transformers to human episodic memory. *Advances in neural information processing systems*, 37:6180–6212, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Rupesh Raj Karn. Linear feedback control systems for iterative prompt optimization in large language models. *arXiv preprint arXiv:2501.11979*, 2025.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. *URL https://arxiv. org/abs/2310.03714*, 3, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Yanbang Li, Ziyang Gong, Haoyang Li, Xiaoqi Huang, Haolan Kang, Guangping Bai, and Xianzheng Ma. Robotic visual instruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12155–12165, 2025.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023a.

Fuxiao Liu, Paiheng Xu, Zongxia Li, Yue Feng, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *arXiv preprint arXiv:2307.05052*, 2023b.

Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025.

Yunong Liu, Cristobal Eyzaguirre, Manling Li, Shubh Khanna, Juan Carlos Niebles, Vineeth Ravi, Saumitra Mishra, Weiyu Liu, and Jiajun Wu. Ikea manuals at work: 4d grounding of assembly instructions on internet videos. *Advances in Neural Information Processing Systems*, 2024.

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

Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pp. 85–99, 2024.

Aengus Lynch, Gbètondji JS Dovonon, Jean Kaddour, and Ricardo Silva. Spawrious: A benchmark for fine control of spurious correlation biases. *arXiv preprint arXiv:2303.05470*, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arxiv. *arXiv preprint arXiv:1703.09312*, 10, 2017.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4040–4048, 2016.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 11048–11064, 2022.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback, 2022. *URL https://arxiv. org/abs/2112.09332*, 35, 2022.

OpenClaw. Openclaw. `https://docs.openclaw.ai/`, 2026. Accessed: 2026-05-13.

Serdar Ozsoy, Shadi Hamdan, Sercan Arik, Deniz Yuret, and Alper Erdogan. Self-supervised learning with an information maximization criterion. *Advances in Neural Information Processing Systems*, 35:35240–35253, 2022.

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models, 2024. *URL https://arxiv. org/abs/2407.07263*, 2022.

Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. Generating physically stable and buildable brick structures from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14798–14809, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhu Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 3804–3814, 2025.

John R Rice. The algorithm selection problem. In *Advances in computers*, volume 15, pp. 65–118. Elsevier, 1976.

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

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2655–2671, 2022.

Ali Safaya and Deniz Yuret. Neurocache: Efficient vector retrieval for long-range language modeling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 870–883, 2024.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 1, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2024.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 86–96, 2016.

Dhruv Shah and Sergey Levine. Viking: Vision-based kilometer-scale navigation with geographic hints. *arXiv preprint arXiv:2202.11271*, 2022.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics: Results of the 11th international conference*, pp. 621–635. Springer, 2017.

Qian Shao, Jiangrui Kang, Qiyuan Chen, Zepeng Li, Hongxia Xu, Yiwen Cao, Jiajuan Liang, and Jian Wu. Enhancing semi-supervised learning via representative and diverse sample selection. *Advances in Neural Information Processing Systems*, 37:111199–111226, 2024.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. *URL https://arxiv.org/abs/2303.11366*, 8, 2024.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pp. 785–799. PMLR, 2023.

Maissa Abir Smaili, Eren Sadikoglu, and Ransalu Senanayake. Memor-e: In-context and fine-tuned llm personalization for alzheimer's assistive robotics. *arXiv preprint arXiv:2605.23941*, 2026.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. Dolma: An open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15725–15788, 2024.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Haoyuan Sun, Jiaqi Wu, Bo Xia, Yifu Luo, Yifei Zhao, Kai Qin, Xufei Lv, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Reinforcement fine-tuning powers reasoning capability of multimodal large language models. *arXiv preprint arXiv:2505.18536*, 2025.

Qilin Sun, Weiming Zhi, Tianyi Zhang, and Matthew Johnson-Roberson. Teaching robots where to go and how to act with human sketches via spatial diagrammatic instructions. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1094–1100. IEEE, 2024.

Priya Sundaresan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. In *8th Annual Conference on Robot Learning*, 2024.

Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9): 2917–2931, 2019.

Stephen Tian, Blake Wulfe, Kyle Sargent, Katherine Liu, Sergey Zakharov, Vitor Guizilini, and Jiajun Wu. View-invariant policy learning via zero-shot novel view synthesis. *arXiv preprint arXiv:2409.03685*, 2024.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Sreevishakh Vasudevan, Som Sagar, and Ransalu Senanayake. Strategic vantage selection for learning viewpoint-agnostic manipulation policies. *arXiv preprint arXiv:2506.12261*, 2025.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems*, 36:74530–74543, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023b.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 15625–15636, 2023c.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 6382–6388, 2019.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yanran Wu, Inez Hua, and Yi Ding. Unveiling environmental impacts of large language model serving: A functional unit view. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10560–10576, 2025.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. k nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. *arXiv preprint arXiv:2303.13824*, 2023.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Wenqian Ye, Luyang Jiang, Eric Xie, Guangtao Zheng, Yunsheng Ma, Xu Cao, Dongliang Guo, Daiqing Qi, Zeyu He, Yijun Tian, et al. The clever hans mirage: A comprehensive survey on spurious correlations in machine learning. *arXiv preprint arXiv:2402.12715*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9134–9148, 2022.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36:19769–19782, 2023.

A Appendix

Table 4: Comparative synthesis of instructional strategies. This table summarizes the underlying principle, applicability, tradeoffs, limitations, and open challenges associated with each instructional form.

Dimension	**Structural**	**In-Context**	**Knowledge-Guided**	**Spatial**	**Temporal**	**Global**	**Hybrid**
Core Principle	Explicitly constrain reasoning trajectories	Demonstrate desired behavior through examples	Ground outputs in external evidence	Encode intent through geometry and visual structure	Explicitly model ordered procedures and transitions	Inject long-range context unavailable locally	Combine multiple instructional signals
Works Best When	Multi-step reasoning, planning, mathematical tasks	Task conventions can be shown through examples, rubrics, or schemas	Factual grounding, source access, or fresh information is required	Robotics, manipulation, navigation, visual reasoning	Long-horizon and procedural tasks	Navigation, planning, structured domains	Real-world multi-constraint tasks
Common Failure Modes	Error propagation, reasoning loops, search explosion	Poor example selection, context overflow	Retrieval failures, stale knowledge, conflicting sources	Ambiguous markings, perception failures	Cascading errors, subtask drift	Incorrect global priors, context dilution	Coordination failures between signals
Data Requirements	None to low	Few representative examples	External corpus or tools	Images, masks, sketches, depth	Demonstrations, manuals, trajectories	Maps, graphs, global structure	Multiple modalities
Interpretability	High	High	High	Very High	High	Moderate	Moderate
Robustness	Moderate	Moderate	High	High	High	High	Very High
Cost	Medium–High	Low	Medium	Low–Medium	Medium	Medium	High
Typical Applications	Question answering, planning, agents, theorem proving	Classification, extraction, style transfer	RAG, tool use, memory systems	VLMs, robotics, navigation	Robotics, workflows, assembly, agents	Navigation, protein modeling, world models	Agentic systems, embodied AI
Open Challenges	Automated discovery of reasoning structures	Example selection, retrieval, and compression	Trust, retrieval quality, and source attribution	Learning from sparse human spatial cues	Automatic task decomposition and verification	Scalable global representations	Automated instruction composition

Table 5: Key literature and concepts categorized by instructional type.

Type	Keywords	References
Classic	Scaling laws & data limits	(Kaplan et al., 2020; Hoffmann et al., 2022; Bahri et al., 2024; Villalobos et al., 2024)
	Stochastic & affine transformations	(LeCun et al., 1998; Yang et al., 2022; Cubuk et al., 2019; Krizhevsky et al., 2012; Takahashi et al., 2019; Howard, 2013)
	NLP augmentation	(Zhang et al., 2015; Wei & Zou, 2019; Sennrich et al., 2016; Feng et al., 2021)
	Physics-based simulators & synthetic data	(Mayer et al., 2016; James et al., 2019; Shah et al., 2017)
	Generative augmentation	(Ho et al., 2020; Rombach et al., 2022; Ren et al., 2025; Chen et al., 2024; Tian et al., 2024)
	Curation, filtering, active selection & DataComp	(Sorscher et al., 2022; Coleman et al., 2019; Gadre et al., 2023; Soldaini et al., 2024; Shao et al., 2024; Vasudevan et al., 2025)
Structural	Prompt engineering & chain-of-thought	(Sahoo et al., 2024; Wei et al., 2022; Kojima et al., 2022; Hase et al., 2024)
	Tree/graph-of-thought, branching search	(Yao et al., 2023; Besta et al., 2024)
	Self-consistency, ReAct, think-act-observe loops	(Wang et al., 2022; Yao et al., 2022)
	Reflexion, self-refine, iterative critique	(Shinn et al., 2024; Madaan et al., 2023; Karn, 2025)
In-Context	Few-shot learning & exemplars	(Brown et al., 2020)
	Rubrics & schema-driven prompts	(Hashemi et al., 2024; Min et al., 2022; Liu et al., 2023b)
	Example retrieval & active example selection	(Xu et al., 2023; Rubin et al., 2022; Zhang et al., 2022)
	Program-of-thought & tool-augmented exemplars	(Chen et al., 2023; Zhou et al., 2022)
	Constitutional rubrics & self-critique guidance	(Bai et al., 2022; Gou et al., 2024)
Knowl.	Retrieval-augmented generation (RAG)	(Lewis et al., 2020)
	Long-term / episodic memory	(Wang et al., 2023a; Fountas et al., 2025; Ji-An et al., 2024)
	kNN caches & experience retrieval	(Safaya & Yuret, 2024)
	Tool-mediated retrieval & just-in-time data acquisition	(Schick et al., 2023; Nakano et al., 2022)
Spatial	Sketches, diagrams, arrows & target regions	(Sundaresan et al., 2024; Sun et al., 2024; Li et al., 2025)
	Segmentation masks & visual grounding	(Long et al., 2015; He et al., 2017; Csurka et al., 2022; Hafiz & Bhat, 2020; Zou et al., 2023; Acuna et al., 2018)
	Depth as geometric instruction	(Chen et al., 2018; Jacquet & Alexis, 2024; Mahler et al., 2017)
	Object enumeration & set-of-mark prompting	(Yang et al., 2023)
	Grid / waypoint / trajectory-based prompting	(Wang et al., 2023c; Huang et al., 2023; Shridhar et al., 2023)
	Information removal & runtime observation intervention	(Hancock et al., 2025)

Continued on next page

Type	Keywords	References
Temporal	Procedural manuals & 4D alignment	(Liu et al., 2024)
	Symbolic build sequences & assembly planning	(Pun et al., 2025)
	Transition-aware representations & intermediate states	(Liu et al., 2024; Pun et al., 2025)
	Stepwise verification checkpoints & procedural supervision	(Liu et al., 2024)
Global	Geographic hints & large-scale navigation priors	(Shah & Levine, 2022)
	Global-local mutual information maximization	(Ozsoy et al., 2022)
	Topology-aware planning & long-horizon coherence	(Shah & Levine, 2022; Ozsoy et al., 2022)
Hybrid	Egocentric sensing & multimodal adaptation	(Liu et al., 2025; Hong et al., 2025)
	Compositional instruction stacks across modalities	(Liu et al., 2025; Hong et al., 2025)