Speculative Decoding for Multimodal Models: A Survey

Anonymous authors
Paper under double-blind review

Abstract

Multimodal generative models have demonstrated remarkable capabilities across diverse domains, from visual understanding and image generation to video processing, audio synthesis, and embodied control. These capabilities, however, incur substantial inference overhead due to autoregressive decoding or iterative generation, which is further compounded by modality-specific challenges such as extensive visual token redundancy, strict real-time latency constraints in robotic control, and prolonged sequential generation in text-to-image synthesis. Speculative decoding has emerged as a promising paradigm to accelerate inference without degrading output quality, yet existing surveys remain focused on text-only large language models. In this survey, we provide a systematic and comprehensive review of speculative decoding methods for multimodal models, spanning Vision–Language, Vision–Language–Action, Video–Language, Speech, Text-to-Image (Vision Auto-Regressive), and Diffusion models. We organize the literature into a unified taxonomy with two primary axes, covering the *draft generation stage* and the *verification and acceptance stage*, complemented by an analysis of inference framework support. Through this taxonomy, we identify recurring cross-modal design patterns, including token compression, KV cache optimization, target-informed transfer, drafter-target alignment, verification cost reduction, relaxed acceptance, and verify-to-draft feedback, and examine how successful techniques transfer across modalities. We further provide a systematic comparison of existing methods under both self-reported and standardized benchmarking settings. Finally, we discuss open challenges and outline future directions. We hope this survey can serve as a valuable resource for researchers and practitioners working on accelerating multimodal inference.

1 Introduction

The sequential nature of autoregressive generation and the iterative nature of diffusion processes severely limit the practical deployment of multimodal generative models, despite their high fidelity in complex reasoning and synthesis tasks. High-resolution images, long video streams, high-frequency robotic control loops, streaming audio codecs, and iterative diffusion trajectories amplify the standard LLM memory-bandwidth bottleneck into a multimodal scaling wall. Visual token sequences exceeding 1000 tokens lead to rapid KV cache growth and substantial prefill costs; Vision–Language–Action (VLA) models face strict real-time control latency limits; speech models must meet stringent time-to-first-audio requirements under multi-codebook generation; Text-to-Image (Vision Auto-Regressive, T2I) models suffer from prolonged token-by-token generation times; and Diffusion Transformers incur high per-step compute across many iterative denoising steps. Together, these factors make sequential and iterative inference latency a primary obstacle to practical multimodal deployment.

Speculative decoding accelerates autoregressive generation without degrading output quality (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2024). A lightweight *draft* model efficiently proposes K candidate tokens, and a full-capacity *target* model verifies these tokens in parallel. This paradigm shifts the computational burden from memory-bandwidth-bound sequential generation to compute-bound batch verification.

Despite the rapid proliferation of speculative decoding techniques, no existing survey addresses their application beyond text-only LLMs (Xia et al., 2024). Extending speculative decoding to multimodal models is not a straightforward application of text-based methods. Multimodal speculation requires specialized drafting

architectures that handle large cross-modal contexts and multi-scale generation. It also demands new verification criteria that relax strict probabilistic matching in favor of feature-level thresholds, phrase-level semantics, perceptual tolerance, and continuous-space coupling. As shown in Figure 1, innovations in Vision–Language Models (VLMs), Vision–Language–Action (VLA) agents, Video–Language models, Speech systems, T2I (Vision AR) generators, and Diffusion-based generators are siloed within their respective sub-communities, lacking a unified framework. Figure 2 contrasts the standard text-only speculative decoding pipeline with the architectures adopted in each multimodal domain, highlighting the domain-specific adaptations required for drafting and verification.

The goal of this survey is to provide a unified overview of speculative decoding for multimodal models. As illustrated in Figure 3, we organize the literature around two primary axes, the *draft generation stage* and the *verification and acceptance stage*, and complement them with an analysis of inference framework support. Together, these components address distinct yet interconnected research questions and provide a systematic view of multimodal speculative decoding. Specifically,

- **Draft Generation (§3):** The draft generation stage determines how candidate tokens are produced efficiently. We survey methods along three dimensions: draft architecture (independent drafters, shared-backbone drafters, and drafter-free speculation), draft execution strategies (multi-token expansion and multi-candidate branching), and draft optimization techniques related to token compression, KV cache optimization, target-informed transfer, and drafter-target alignment.
- **Verification and Acceptance (§4):** The verification stage determines how drafted tokens are validated against the target model. We survey verification execution strategies (linear, tree-based, path-level, and iterative verification) and optimization techniques including cost reduction, relaxed acceptance criteria, and verify-to-draft feedback loops.
- **Inference Frameworks (§5):** We survey existing frameworks that provide system-level support for speculative decoding, covering their unique features and optimizations for multimodal workloads.

We further provide a systematic comparison of existing methods under both self-reported and standardized benchmarking settings (§6), and discuss open challenges to outline future directions (§7).

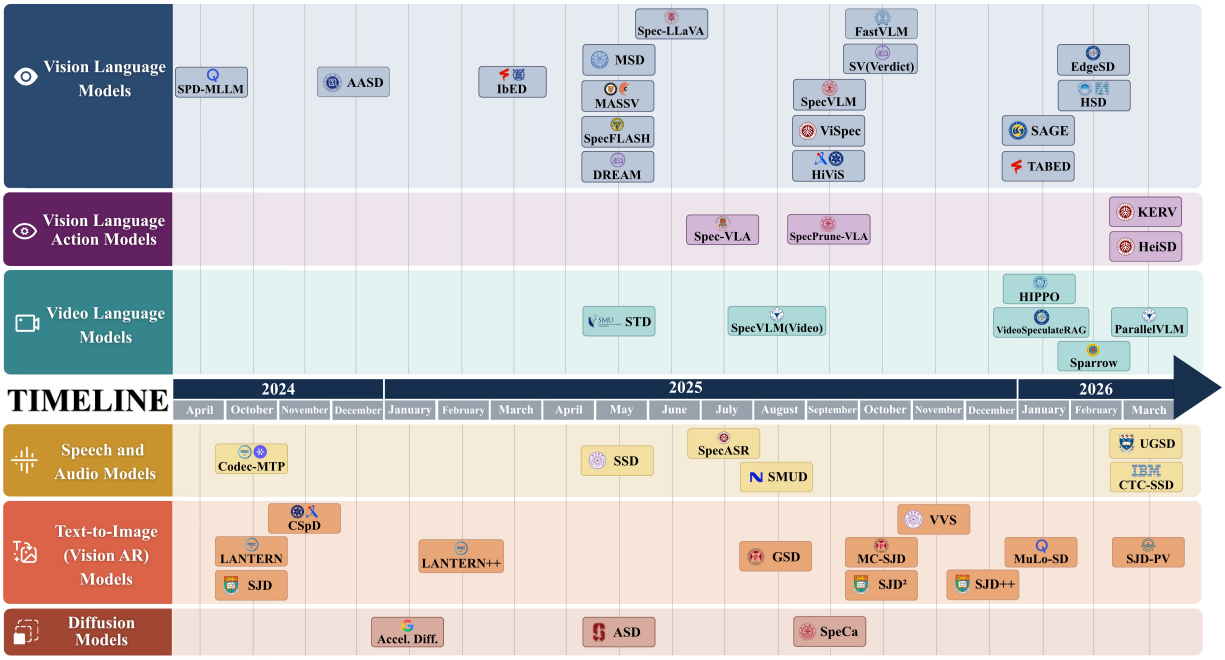

Figure 1: Timeline of multimodal speculative decoding methods surveyed in this work. Colors indicate modality: methods span Vision–Language, Vision–Language–Action, Video–Language, Speech, T2I (Vision AR), and Diffusion models.

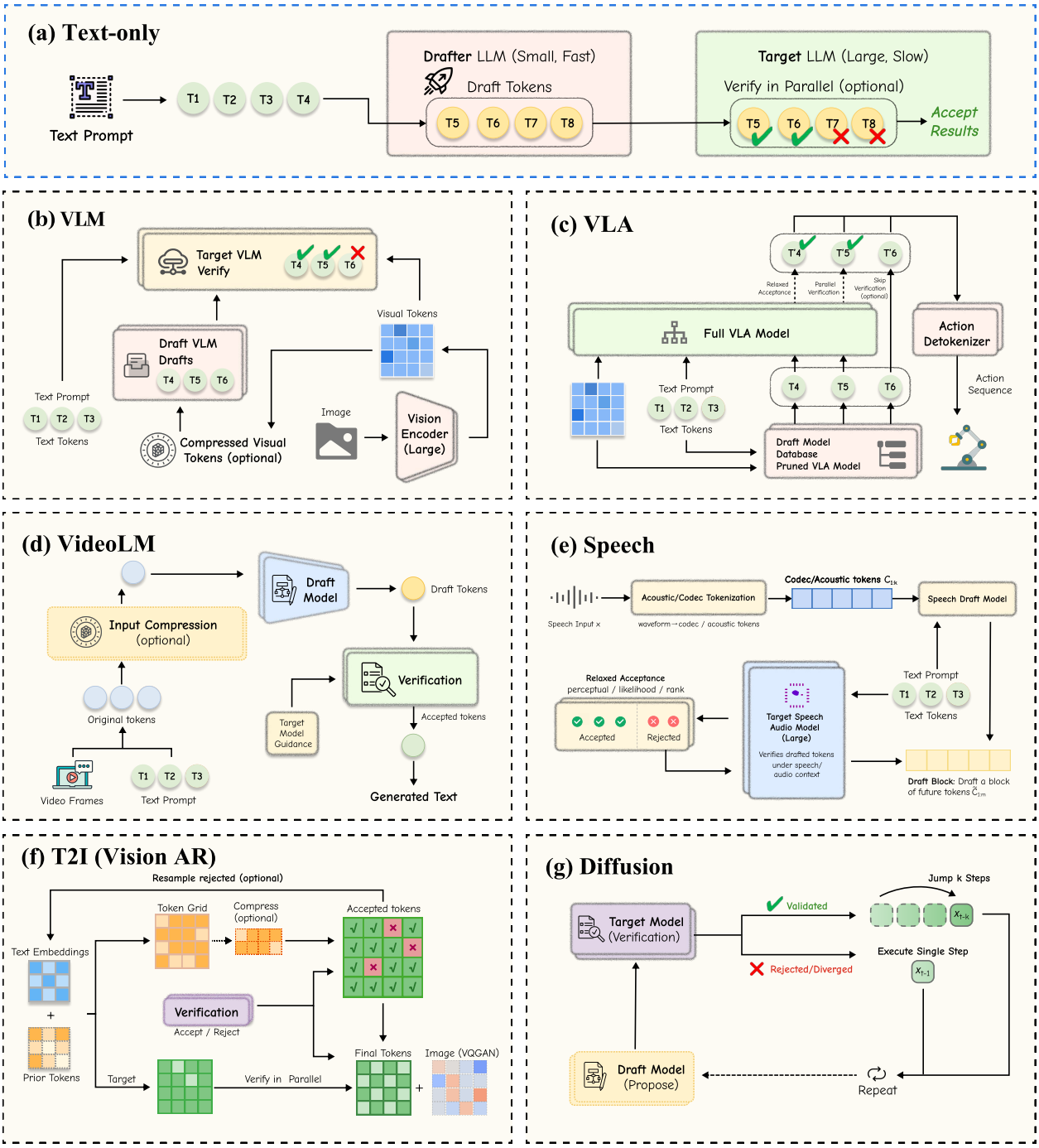

Figure 2: Overview of speculative decoding architectures across modalities. (a) **Text-only**: a small LLM drafts tokens verified by a large LLM. (b) **VLM**: drafting conditioned on text and (optionally compressed) visual tokens; the target VLM verifies with full visual encoding. (c) **VLA**: a compact or pruned VLA drafts action tokens, verified by the full VLA model. (d) **VideoLM**: video frames are tokenized with optional compression; the drafter generates tokens guided by target features. (e) **Speech**: acoustic inputs are tokenized via neural codecs; the target verifies with relaxed acceptance criteria. (f) **T2I (Vision AR)**: visual tokens are drafted and verified in parallel; accepted tokens are decoded into images via a visual decoder (e.g., VQGAN). (g) **Diffusion**: a draft model proposes multi-step trajectory jumps, while the target model validates the proposed trajectory and rolls back upon divergence.

2 Background

2.1 Standard Speculative Decoding

Standard speculative decoding accelerates autoregressive inference by decomposing generation into a *draft-then-verify* paradigm (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2024). Given a context $x_{<t}$, a lightweight draft model $\mathcal{M}_{\text{draft}}$ with distribution $p_{\text{draft}}(x \mid x_{<t})$ autoregressively generates K candidate tokens $\tilde{x}_t, \ldots, \tilde{x}_{t+K-1}$. The full-capacity target model $\mathcal{M}_{\text{target}}$ with distribution $p_{\text{target}}(x \mid x_{<t})$ then verifies all K candidates in a single parallel forward pass. Each candidate token $\tilde{x}_{t+i}$ is accepted with probability:

$$\alpha_{t+i} = \min\left(1,\ \frac{p_{\text{target}}(\tilde{x}_{t+i} \mid x_{<t+i})}{p_{\text{draft}}(\tilde{x}_{t+i} \mid x_{<t+i})}\right). \tag{1}$$

The first rejected token truncates the drafted sequence. The target model then resamples a token from a modified distribution to correct the error, and the process repeats. This exact-match acceptance criterion ensures that the final output distribution matches $p_{\text{target}}(x \mid x_{<t})$ exactly.

The expected speedup hinges on the token acceptance rate $\mathbb{E}[\alpha]$ and the computational overhead of generating K tokens via $\mathcal{M}_{\text{draft}}$. While effective for text, applying this exact discrete formulation to multimodal contexts often yields suboptimal $\mathbb{E}[\alpha]$ due to domain mismatches, necessitating the specialized mechanisms categorized in this survey.

2.2 Multimodal Generation Architectures

We analyze the distinct architectures and computational bottlenecks across six multimodal domains.

Vision–Language Models (VLMs) fuse pre-trained visual encoders (e.g., ViT (Dosovitskiy et al., 2020)) with large language models (Liu et al., 2023; Team, 2023). High-resolution images are tokenized into long sequences (often 1000+ tokens per image), causing substantial prefill overhead and rapid memory exhaustion during the autoregressive phase. Visual token redundancy offers unique opportunities for draft compression.

Vision–Language–Action Models (VLAs) map visual input and textual instructions directly to low-level robotic control tokens (Zitkovich et al., 2023). Operating in physical environments imposes hard, real-time inference latency limits that pure autoregressive decoding struggles to meet.

Video–Language Models (VideoLMs) extend VLMs across the temporal dimension, processing streams of frame-level visual tokens (Lin et al., 2024). Token count scales linearly with video length, exacerbating Key-Value (KV) cache bottlenecks and introducing a requirement for temporal coherence during drafting.

Speech and Audio Models generate discretized audio using neural audio codecs representing waveforms at multiple quantization levels (Défossez et al., 2022). Because human hearing processes audio semantically, these models exhibit *perceptual tolerance*: acoustically equivalent sounds can have differing discrete representations, violating the premise of Eq. 1 and requiring relaxed verification criteria.

Text-to-Image (Vision Auto-Regressive, T2I) Models synthesize images by predicting discrete codebook indices within an autoregressive transformer (Esser et al., 2021). These models suffer from prolonged generation times due to long flattened token sequences, motivating speculative acceleration of the sequential token prediction process.

Diffusion Models generate structured outputs by iteratively removing noise from continuous latent spaces (Ho et al., 2020; Song et al., 2020). Because they do not operate on discrete vocabularies or standard left-to-right decoding, they require distinct "proposal-and-verification" paradigms framed around trajectory dynamics rather than token-level probabilities. Diffusion is not naturally compatible with prefix-wise verification: it operates over continuous trajectories without a discrete prefix structure, making partial correctness and local rollback ill-defined. Diffusion models are also widely used for text-to-image generation

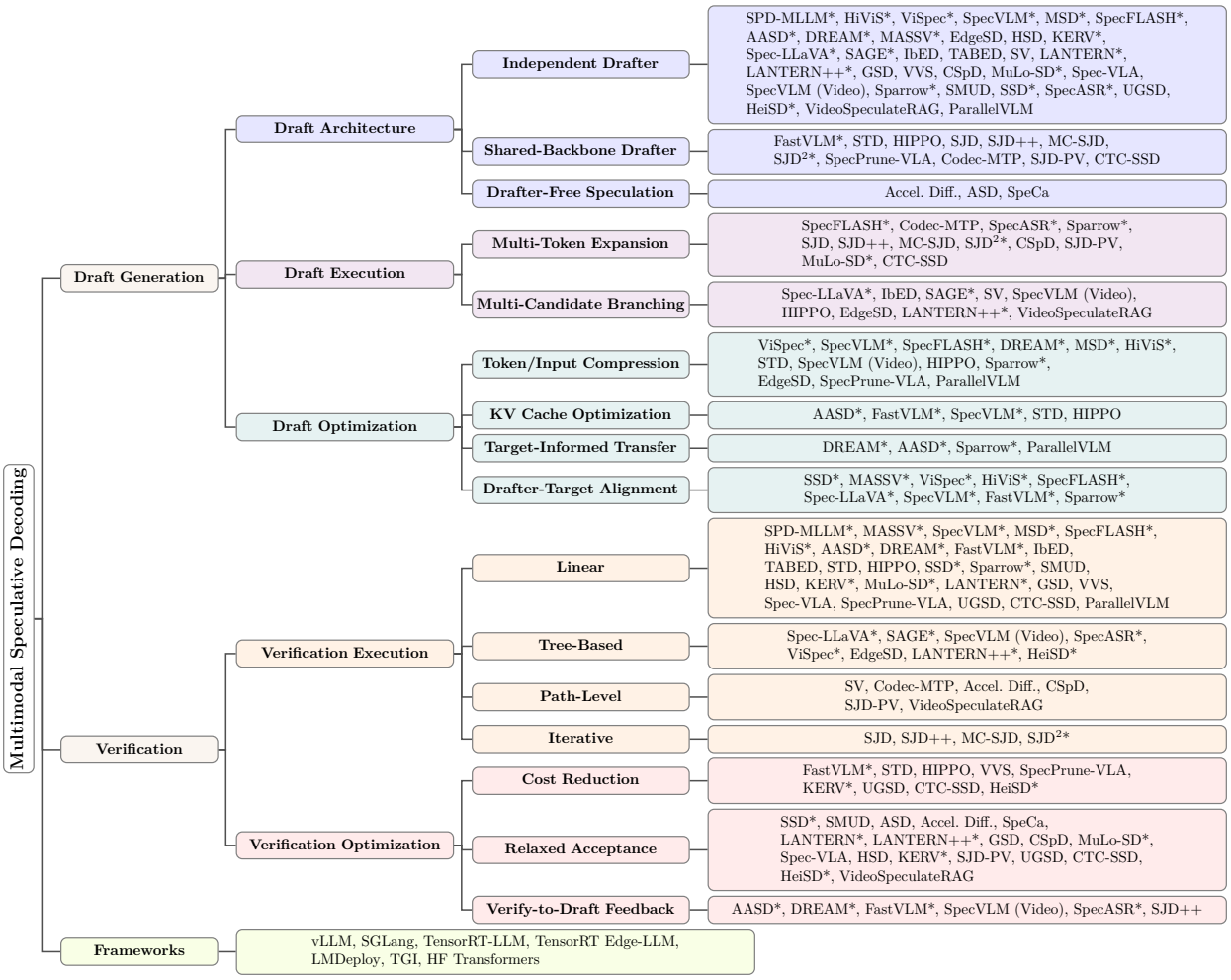

Figure 3: Taxonomy of speculative decoding for multimodal models. * denotes methods requiring training/fine-tuning; unmarked methods are training-free.

Blue : draft architecture. Violet : draft execution. Teal : draft optimization. Orange : verification execution. Red : verification optimization. Green : frameworks.

through latent diffusion; however, because their speculative decoding mechanisms differ fundamentally from autoregressive approaches, they are treated as a separate category in this survey.

3 Draft Generation Stage

The draft generation stage directly determines overall speedup through two factors: the *speculation accuracy* of the drafter, measured by the average number of accepted tokens per step, and *drafting latency*. Balancing high speculation accuracy against low drafting latency is particularly challenging in the multimodal setting, as each modality introduces distinct constraints on candidate generation.

In this section, we classify various drafting strategies following the taxonomy in Figure 3 into three dimensions: draft architecture (§3.1), draft execution (§3.2), and draft optimization (§3.3).

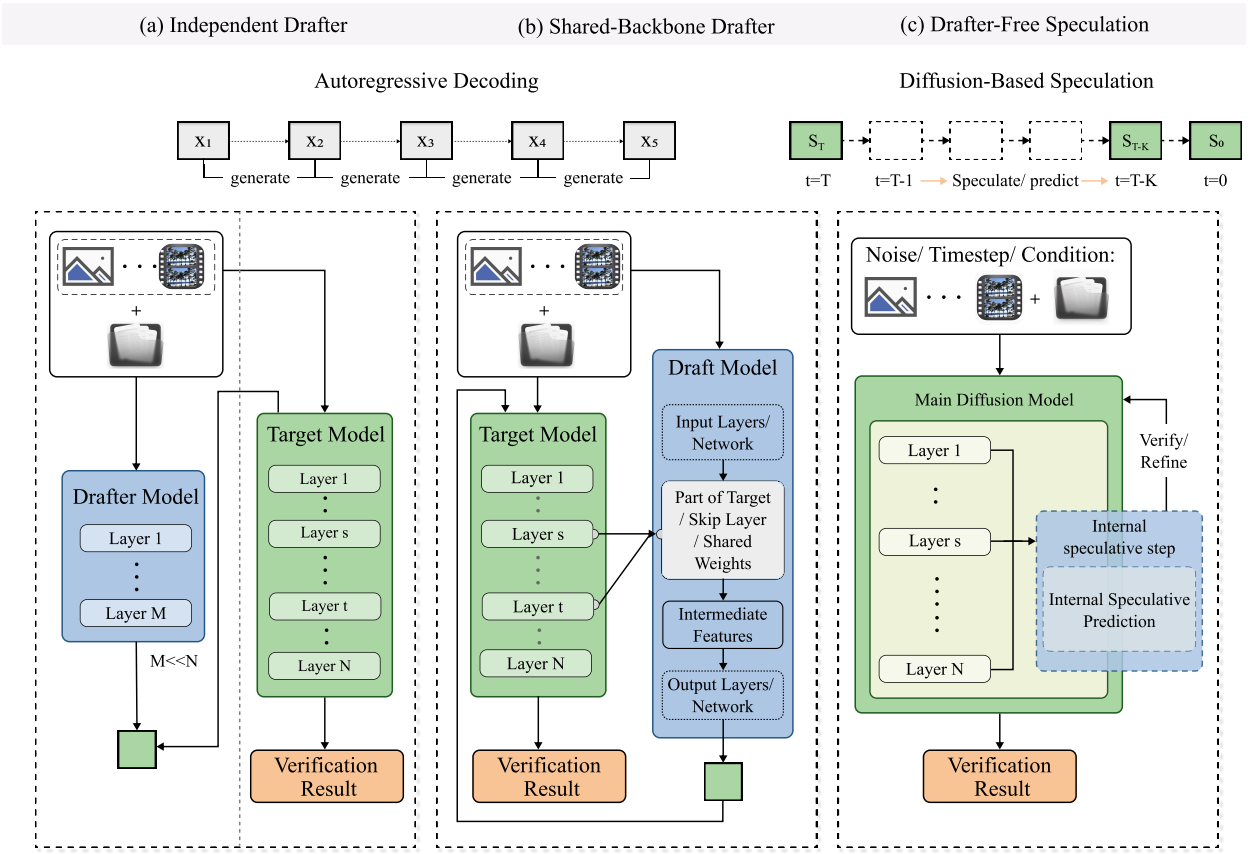

Figure 4: Architectural overview of the three draft architecture categories. (a) **Independent Drafter**: a separate, smaller model generates draft tokens that the target verifies. (b) **Shared-Backbone Drafter**: the target model itself serves dual roles, using a sub-graph for fast drafting and the full model for verification. (c) **Drafter-Free Speculation**: applicable to diffusion models, where mathematical properties of the denoising process enable speculative steps without any auxiliary model.

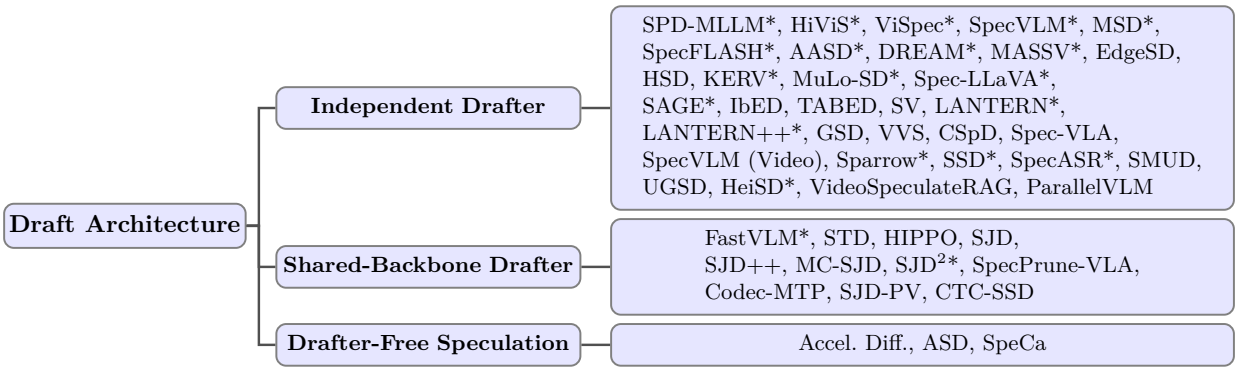

Figure 5: Sub-taxonomy of draft architecture (§3.1). Methods are categorized by their structural relationship to the target model. * denotes methods requiring training/fine-tuning.

3.1 Draft Architecture

Figure 4 illustrates the structural overview of these categories, Figure 5 presents the sub-taxonomy of draft architecture, and Table 1 summarizes their formulations.

Methods	Formulation ($p_1 \dots p_K$ or $\tilde{x}_{1\dots K}$)	Architecture
Independent Drafters	$p_{1\dots K} = \mathcal{M}_{\text{draft}}(x \mid \text{prompt})$ *(Separate model)*	Small VLM / LM, Vision Predictor, ConvNet Head, Retrieval-Based
Shared Backbone	$p_{1\dots K} = \mathcal{M}_{\text{target}}(x \mid \text{prompt}, \text{skip})$ *(Target sub-graph)*	Early Exiting, Layer Skipping, Sparse KV Routing, Jacobi Self-Drafting
Drafter-Free Speculation	$\tilde{z}_{1\dots K} \sim \mathcal{F}_{\text{proposal}}(s_t)$ *(No auxiliary model)*	Coupling Jumps, Exchangeability, Feature Forecasting

Table 1: Summary of formulations for draft architecture in multimodal speculative decoding. $p_{1\dots K}$ denotes the draft probability outputs for K candidate positions, $\mathcal{M}_{\text{draft}}$ and $\mathcal{M}_{\text{target}}$ are the draft and target models (defined in §2.1), $\tilde{z}_{1\dots K}$ are proposed latent states, $\mathcal{F}_{\text{proposal}}$ is a training-free proposal function, and s_t is the diffusion state at step t. Methods are categorized by their structural relationship to the target model.

3.1.1 Independent Drafter

Independent drafters balance speculation accuracy against drafting efficiency by decoupling the draft model from the target. In the multimodal setting, the key design question is how the drafter handles visual and other non-text inputs. We categorize independent drafters by their treatment of these modalities.

Text-Only Independent Drafters. Text-only drafters discard visual and acoustic inputs during drafting entirely, simplifying the drafting distribution to $p_{\text{draft}}(y \mid x_{\text{text}})$, where y denotes the output token and x_{text} the textual input.

SPD-MLLM (Gagrani et al., 2024) provides the foundational feasibility proof: applying speculative decoding to a VLM with a language-only drafter achieves meaningful speedup, demonstrating that the draft stage need not process image tokens at all, since only the target must verify correctness.

Vision–Language Independent Drafters. To recover the acceptance rate degradation caused by discarding non-text inputs, this family of drafters explicitly processes multimodal tokens, modeling $p_{\text{draft}}(y \mid x_{\text{text}}, x_{\text{vision}})$. Design choices range from lightweight visual adaptors to full multimodal projectors, and from purely independent forward passes to architectures that receive precomputed features from the target to avoid redundant visual encoding.

ViSpec (Kang et al., 2025) introduces a Q-Former-style (Li et al., 2023) vision adaptor that compresses visual features into a small set of query tokens and a global visual vector injected at every text position. SpecVLM (Huang et al., 2025) addresses the prefill bottleneck through an elastic visual compressor supporting multiple adaptive compression modes. HiViS (Xie et al., 2025) removes explicit visual token processing from the drafter entirely, instead conditioning on a precomputed semantic embedding exported by the target model. SpecFLASH (Wang et al., 2025c) combines latent-aware visual token compression with semi-autoregressive drafting, allowing it to propose blocks of candidate tokens at once. MSD (Lin et al., 2025a) decouples text and vision in the drafting pipeline, processing each modality according to its characteristics, while MASSV (Ganesan et al., 2025) trains a compact independent LM equipped with a lightweight multimodal projector. AASD (Yang et al., 2025) employs a dedicated speculative module that reuses the target's KV cache via learned projections. DREAM (Hu et al., 2025b) conditions the drafter on target intermediate features via adaptive cross-attention. Beyond single-drafter designs, IbED (Lee et al., 2025) (In-batch Ensemble Drafting) runs multiple independent strategies as a batch, and TABED (Lee et al., 2026) extends this with training-free test-time adaptation that selects ensemble weights minimizing KL divergence against the target distribution. EdgeSD (Huang et al., 2026) introduces a *vision-decoding disaggregation* (VED) architecture for edge-cloud VLM speculative decoding, decoupling the visual encoder and LLM backbone across separate edge servers; it further contributes bandwidth-aware dynamic image token merging and an adaptive token tree

via delta-stepping. HSD (Liao et al., 2026) (Hierarchical Speculative Decoding) targets the document parsing scenario, where VLMs must generate long structured outputs (e.g., full-page Markdown): a lightweight pipeline-based parser serves as the draft model, the page is partitioned into semantically independent regions whose draft–verify cycles execute in parallel, and a second page-level verification stage reassembles all accepted region outputs against the full VLM with a tolerance-based acceptance criterion (τ-matching).

Vision–Language–Action Independent Drafters. SPEC-VLA (Wang et al., 2025b) applies speculative decoding to Vision–Language–Action models by pairing a compact VLA drafter with the full-scale target policy; the drafter generates candidate action tokens autoregressively, which the target verifies under a relaxed acceptance criterion that tolerates functionally equivalent control signals. KERV (Zheng et al., 2026a) (Kinematic-Rectified Speculative Decoding) further advances VLA speculation by combining token-domain drafting with kinematic-domain compensation: when verification rejects a draft token, KERV activates a Kalman Filter that predicts the remaining actions from the robot's kinematic history, avoiding GPU-side recomputation entirely. HEISD (Zheng et al., 2026b) introduces *hybrid* speculative decoding for VLA models, dynamically switching between retrieval-based and drafter-based SD according to trajectory kinematics. A kinematic-based fused metric combining curvature radius and cumulative displacement determines the hybrid boundary: high-metric (straight, fast-moving) segments use retrieval-based SD for near-zero drafting overhead, while low-metric (curved, fine-grained) segments fall back to a trained drafter. To make retrieval-based SD practical, HEISD introduces an adaptive verify-skip mechanism that selectively bypasses verification when feature similarity to historical trajectories is high, and a sequence-wise relaxed acceptance strategy that groups kinematically correlated tokens and accepts the entire sequence when its aggregate bias remains small.

Video–Language Independent Drafters. SPARROW (Zhang et al., 2026) targets long-video scenarios where extensive visual token sequences cause attention dilution, offloading visual computation to the target model and eliminating the drafter's visual KV cache through target-informed knowledge transfer and attention constraints (detailed in §3.3). To handle the multi-document retrieval setting, VIDEOSPECULATERAG (Li & Liu, 2026) introduces speculative decoding into the video RAG pipeline: each retrieved document is independently paired with the video and question and processed in parallel by a lightweight draft VLM, with a larger verifier scoring each candidate through δ-tolerant reliability filtering followed by entity-alignment reranking. Addressing the orthogonal challenge of pipeline efficiency, PARALLELVLM (Kong et al., 2026) pairs a same-family smaller VLM as a training-free independent drafter with a fully parallel pipeline where prefilling and decoding of draft and target execute concurrently, hiding draft overhead under the target's latency. An Unbiased Verifier-Guided Pruning (UV-Prune) strategy selects draft visual tokens based on vision–text similarity variations across the target model's early layers, avoiding the positional bias of attention-guided pruning.

Speech Independent Drafters. Speech speculative decoding exploits the low-entropy, strongly conditioned nature of acoustic generation. SSD (Lin et al., 2025b) (Speech Speculative Decoding) employs a compact, independently trained audio language model as the drafter, generating candidate codec token sequences that are then verified by the full TTS model. SPECASR (Wei et al., 2025) applies a related paradigm to automatic speech recognition, pairing a lightweight draft ASR model with the full target recognizer. SMUD (Okabe & Yamamoto, 2025) introduces an alternative paradigm: rather than using a separate draft model, a CTC greedy search provides a pseudo-draft by generating a preliminary sequence and masking low-confidence regions. Mask boundaries are then refined via a single non-autoregressive decoder forward pass, acting as an efficient one-shot pseudo-draft step. UGSD (Xue et al., 2026) (Uncertainty-Guided Speculative Decoding) frames speech emotion captioning as an edge–cloud collaborative pipeline: a lightweight SALM drafts captions on-device, and only token blocks whose maximum entropy exceeds a threshold are escalated to a cloud-side LALM verifier. The verifier applies a rank-based acceptance rule, and the block length adapts dynamically based on recent acceptance history.

Text-to-Image (Vision AR) Independent Drafters. LANTERN (Jang et al., 2025a) and LANTERN++ (Jang et al., 2025b) train compact visual AR models as drafters for image generation, retaining the standard dual-model framework while introducing relaxed latent-space acceptance to overcome the low token-match rates associated with visual codebook prediction. GSD (So et al., 2025b) pairs an

independent drafter with dynamic token clustering, grouping visually equivalent tokens to boost acceptance without training. VVS (Dong et al., 2025) and CSpD (Wang et al., 2024) similarly employ independent drafters but shift their innovations to verification skipping and continuous-density acceptance, respectively. MuLo-SD (Peruzzo et al., 2026) (Multi-Scale Local Speculative Decoding) introduces a multi-scale drafting paradigm: a low-resolution AR model generates coarse draft tokens that are expanded to the target resolution via a trained up-sampler, exploiting the natural hierarchy of image resolutions. The verification-side innovations of these T2I methods, including latent-space neighborhood acceptance, grouped acceptance, and local spatial relaxation, are analyzed in §4.2.2.

The optimization strategies underlying these independent drafters, including visual token compression, KV cache sharing, target-informed knowledge transfer, and drafter–target alignment, are discussed later in this section, while their verification-side design choices are covered in Section 4.

3.1.2 Shared-Backbone Drafter

Several methods eliminate the separate draft model by repurposing the target model itself for efficient drafting. This shared-backbone approach reduces the two-model system to a single model that operates at two computational granularities.

In the VLM domain, FastVLM (Bajpai & Hanawal, 2025) eliminates the separate draft model entirely through self-speculative decoding. The first n layers of a single VLM backbone serve as the drafter; the full L-layer model performs verification. Because draft and verification share the same weights and layer ordering, computation from the first n layers transfers directly to the verification pass; an imitation network bridges the representation gap between stages.

In the VLA domain, SpecPrune-VLA (Wang et al., 2025a) implements self-speculative decoding through action-aware visual token pruning: the pruned model serves as its own drafter within a shared backbone, eliminating the need for a separate draft model, a practical advantage for embodied deployment where memory is constrained.

In the VideoLM domain, STD (Zhang et al., 2025) implements a KV-split variant of shared-backbone drafting. Rather than splitting by layer depth, it splits by attention density: the same backbone serves both roles, but drafting uses a sparse subset of attention entries while verification restores the full dense attention. This approach exploits the empirical observation that VideoLM attention is consistently sparse during decoding, making a single backbone sufficient for both fast drafting and accurate verification. Rather than splitting attention, HIPPO (Lv et al., 2026) realizes shared-backbone drafting through pipelined overlapping: it overlaps target verification of batch t with draft generation of batch $t+1$, hiding verification latency behind drafting computation via a double-buffer KV cache management scheme. Its algorithmic contribution, holistic video token scoring that fuses global semantic relevance, temporal redundancy, and spatial redundancy signals, is detailed in §3.3.1.

In the speech domain, Codec-MTP (Nguyen et al., 2025) equips the target model's internal layers with lightweight multi-token prediction heads, enabling it to self-propose blocks of future codec tokens in a single forward pass. CTC-SSD (Saon et al., 2026) (Self-Speculative Decoding) reuses the CTC encoder head of a speech-aware LLM as the drafter: the greedy CTC hypothesis is generated non-autoregressively, and high-confidence outputs are accepted under a relaxed criterion. When verification fails, AR decoding resumes from the accepted prefix. This three-stage pipeline simultaneously improves WER (through complementary CTC-LLM error patterns) and accelerates inference.

In the T2I (Vision AR) domain, the SJD family (Teng et al., 2025b;a; So et al., 2025a; Teng et al., 2025c) applies this self-drafting philosophy to visual autoregressive generation through Jacobi-style parallel prediction. SJD (Teng et al., 2025b) initializes multiple token positions simultaneously and iterates the target AR model in Jacobi mode, accepting tokens that reach probabilistic convergence; no auxiliary model is required. SJD++ (Teng et al., 2025a) reuses high-confidence tokens across iterations to accelerate convergence, while MC-SJD (So et al., 2025a) stabilizes convergence via maximal coupling. SJD-PV (Yu et al., 2026) inherits the Jacobi self-drafting framework, using the target model's iterative refinement to propose candidate tokens which the target then validates via phrase-level joint acceptance. These methods share the backbone

philosophy of FastVLM (Bajpai & Hanawal, 2025) and STD (Zhang et al., 2025) but differ in mechanism: rather than layer-splitting or KV-splitting, they exploit iterative fixed-point convergence of the full model.

3.1.3 Drafter-Free Speculation

In contrast to the preceding categories, drafter-free methods accelerate diffusion inference without constructing a separate draft model or repurposing a sub-graph of the target. Instead, they exploit mathematical properties of the diffusion process itself, including coupling structure, timestep exchangeability, and feature smoothness, to speculate over multiple steps at once. Because no auxiliary model is trained or maintained, these approaches are structurally lightweight, though their applicability remains specific to continuous generative processes.

Accelerated Diffusion Sampling (De Bortoli et al., 2025) constructs training-free proposals via reflection maximal coupling: given the current noisy sample x_t, it proposes a future sample x_{t-k} by exploiting the geometric structure of the SDE, requiring no learned draft model. ASD (Hu et al., 2025a) discovers that diffusion timesteps are exchangeable under stochastic localization, enabling the model to evaluate multiple timestep orderings in parallel without constructing any draft; the target model itself validates the reordered proposals. SpeCa (Liu et al., 2025a) caches intermediate features across denoising steps and uses Taylor expansion to forecast future activations, converting cached computation into speculative proposals that bypass redundant model evaluations.

3.2 Draft Execution

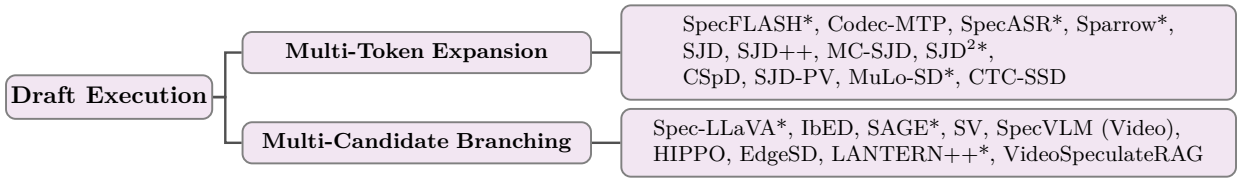

Figure 6: Sub-taxonomy of draft execution strategies (§3.2). Multi-token expansion explores depth; multi-candidate branching explores width. * denotes methods requiring training/fine-tuning.

Beyond the choice of drafting mechanism, the *structure* of the speculation, i.e., how candidates are organized in width and depth, strongly influences both speculation accuracy and the number of tokens decoded per step. The majority of surveyed methods default to standard single-token autoregressive drafting (or, for diffusion models, the standard single-step denoising schedule), generating one candidate token per forward pass prior to verification. The two subcategories below, multi-token expansion and multi-candidate branching, highlight methods that innovate beyond this default paradigm by producing multiple tokens or multiple candidate sequences per drafting round. Conceptually, multi-token expansion increases speculation depth along a single trajectory, while multi-candidate branching expands the width of the candidate space. Figures 6 and 7 illustrate the draft execution sub-taxonomy and strategies, and Table 2 contrasts the two approaches.

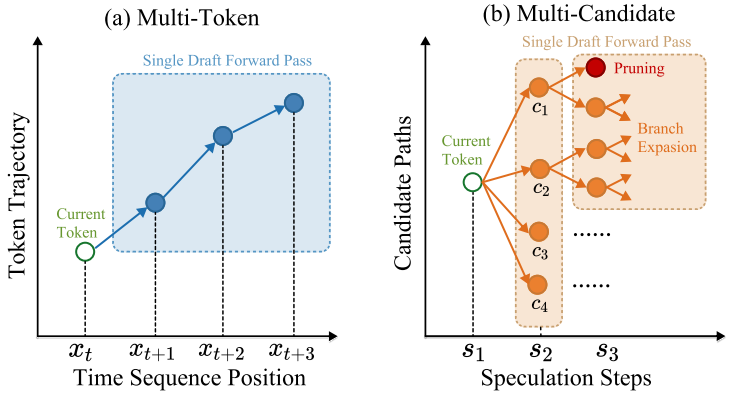

Figure 7: Illustration of two draft execution strategies. (a) **Multi-Token Expansion**: multiple future tokens $(x_{t+1}, x_{t+2}, x_{t+3})$ are predicted in a single draft forward pass along a single trajectory, reducing the number of serial drafting steps. (b) **Multi-Candidate Branching**: multiple candidate paths $(c_1, c_2, \dots)$ are generated simultaneously, forming a speculation tree that is pruned and expanded over successive steps; the target model verifies all branches in parallel.

Strategy	State (Q) & Pattern	Drafter Mechanism
Multi-Token Expansion	$Q = \{t+1, \ldots, t+K\}$ $p_{t+k} = f_k(h_t)$, $k = 1, \ldots, K$	Block Prediction, Jacobi Refinement, Semi-AR Heads
Multi-Candidate Branching	$Q = \{c_1, c_2, \ldots, c_N\}$ $c_n \sim \mathcal{M}_{\text{draft}}(x \mid \eta_n)$	Branch Expansion, Speculation Trees, Parallel Proposals

Table 2: Unified view of draft execution strategies. t denotes the current decoding position, h_t the hidden state at position t, and f_k the prediction function for the k-th future token. c_n is the n-th candidate sequence, η_n its strategy-specific branching parameter (e.g., temperature or prompt variant), and N the total number of candidates. Multi-token expansion predicts multiple future positions along a single trajectory, while multi-candidate branching explores multiple candidate continuations at the same step.

3.2.1 Multi-Token Expansion

Generating multiple future tokens per forward pass reduces the number of serial draft steps required per speculation round.

In the VLM domain, SpecFLASH (Wang et al., 2025c) equips the drafter with semi-autoregressive heads that produce K tokens simultaneously. Placeholder tokens fill positions for not-yet-generated tokens within a block; blocks are decoded autoregressively (each block conditions on the previous block's output) while tokens within each block are generated in parallel. This block-wise decoding reduces the number of serial forward passes by a factor of K. AASD (Yang et al., 2025) and FastVLM (Bajpai & Hanawal, 2025) adopt standard γ-token (i.e., $\gamma = K$ draft tokens per step) speculative decoding for parallel verification, although their primary contributions lie in target–draft interaction and backbone sharing rather than dedicated multi-token head design.

In the VideoLM domain, Sparrow (Zhang et al., 2026) employs a multi-token prediction strategy to bridge the training–inference distribution gap, generating multiple draft tokens per step via recursive self-conditioning.

In the speech domain, depth parallelism is the dominant strategy because ASR and TTS are strongly conditioned, low-entropy generation tasks that favor extensive long-sequence speculation over multi-branch exploration. Codec-MTP (Nguyen et al., 2025) predicts n future codec tokens in a single forward pass, reducing decoding steps proportionally. SpecASR (Wei et al., 2025) adaptively extends draft length, dynamically adjusting based on prediction confidence to minimize verification rounds.

In the T2I (Vision AR) domain, the SJD family (Teng et al., 2025b;a; So et al., 2025a; Teng et al., 2025c) achieves depth parallelism natively through its Jacobi iteration framework: multiple token positions are predicted simultaneously in each forward pass and iteratively refined until convergence, generating blocks of tokens per round without auxiliary prediction heads. SJD^2 (Teng et al., 2025c) further structures each Jacobi round as a denoising trajectory from Gaussian noise. CSpD (Wang et al., 2024) adapts multi-token block prediction to continuous-valued visual AR models, combining denoising trajectory alignment with token pre-filling to construct density-aligned candidate blocks. MuLo-SD (Peruzzo et al., 2026) introduces a multi-scale drafting strategy for visual AR models: a low-resolution drafter generates coarse candidate tokens, which are then up-sampled via a learned up-sampler into the target resolution, producing a full-resolution candidate patch from each low-resolution draft token without increasing draft sequence length.

3.2.2 Multi-Candidate Branching

Several methods expand the candidate space by generating multiple draft branches simultaneously, forming a speculation tree that the target verifies in a single pass.

In the VLM domain, Spec-LLaVA (Huo et al., 2025) constructs a dynamic speculation tree with online structural and budget pruning: structural pruning removes low-probability or redundant branches, while

budget pruning limits the tree to the top-n candidate tokens, preventing tree explosion; the target model then verifies this tree structure. Rather than branching at the token level, SV (Speculative Verdict) (Liu et al., 2025b) operates at the reasoning level: multiple lightweight VLMs independently generate diverse reasoning paths, a consensus filter based on negative log-likelihood scores selects high-agreement paths, and the verdict model synthesizes a new final answer in a single inference call, functioning as an evidence synthesizer rather than a voter. While the above methods use fixed or pre-defined tree structures, SAGE (Tong et al., 2026) dynamically adjusts the tree shape using the drafter's prediction entropy at each step, allocating wider branching for high-entropy (uncertain) predictions and deeper speculation for low-entropy predictions. EdgeSD (Huang et al., 2026) takes adaptivity further by formulating tree generation as a single-source shortest path (SSSP) problem, solved via a parallel delta-stepping algorithm that maximizes the expected number of accepted tokens under strict computational budgets of edge servers.

In the VideoLM domain, SpecVLM (Video) (Ji et al., 2025) adopts an EAGLE-style (Li et al., 2024a) static tree structure with tree attention masks; the draft model generates multi-branch candidate trees over pruned video tokens, which the target verifies via structured tree attention. Instead of token-level trees, VideoSpeculateRAG (Li & Liu, 2026) takes a document-parallel branching approach, generating one complete candidate answer per retrieved document and treating each document–video–question triple as an independent branch; the verifier selects the best branch through path-level two-stage scoring. Finally, HIPPO (Lv et al., 2026) shares the backbone between draft and verification (§3.1.2), but its pipelined overlap of consecutive batches across time steps is more naturally viewed as an execution-level multi-candidate strategy that transforms the serial draft-then-verify cycle into a production-line pipeline.

In the T2I (Vision AR) domain, LANTERN++ (Jang et al., 2025b) applies static tree drafting to visual autoregressive generation, constructing a fixed tree topology that enables multi-branch speculation over visual codebook tokens (§4.2.2).

3.3 Draft Optimization

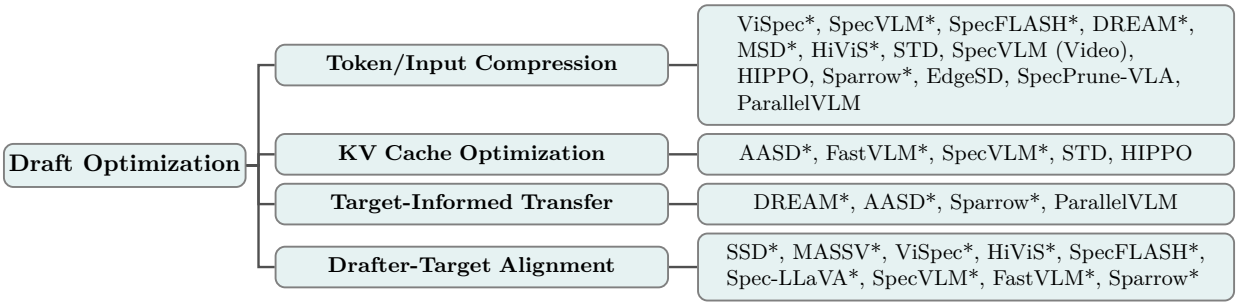

Figure 8: Sub-taxonomy of draft optimization strategies (§3.3). These strategies recur across modalities. Target-Informed Transfer covers inference-time signal injection; Drafter-Target Alignment covers training-time compatibility. * denotes methods requiring training/fine-tuning.

As shown in Figures 8 and 9, four recurring optimization strategies emerge that transcend domain boundaries. These patterns, namely token compression, KV cache optimization, target-informed knowledge transfer, and drafter-target alignment, are observed across vision–language, VLA, video, speech, and T2I (Vision AR) systems, where high-dimensional inputs introduce substantial redundancy and require careful draft–target coordination.

3.3.1 Token/Input Compression

Multimodal inputs are fundamentally redundant; drafters that compress aggressively while targets retain full resolution during verification can substantially reduce drafting latency without sacrificing speculation accuracy.

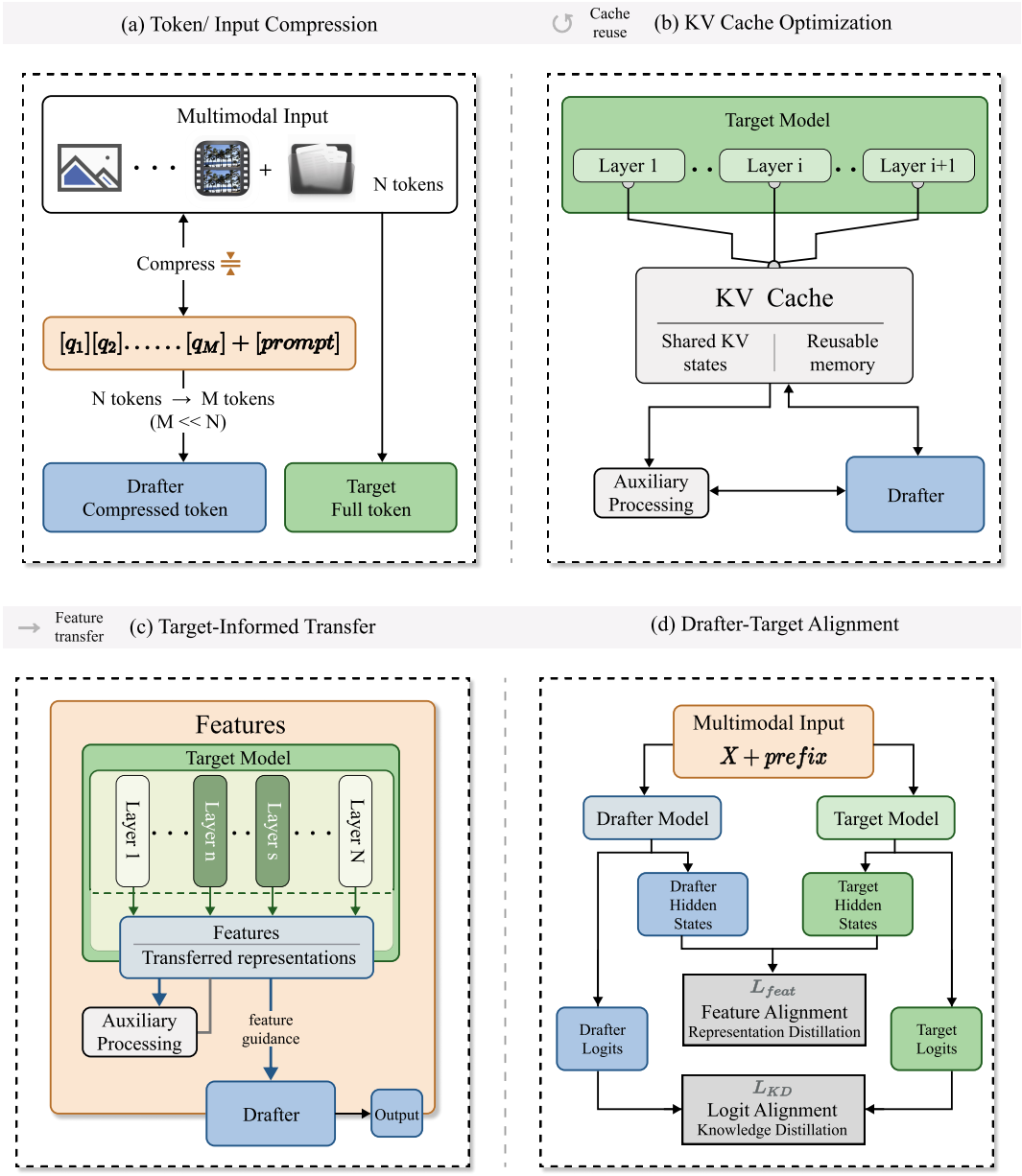

Figure 9: Illustration of four draft optimization strategies in multimodal speculative decoding. (a) **Token/Input Compression**: multimodal inputs are compressed from N tokens to $M \ll N$ tokens via semantic or adaptive compression, reducing drafting latency while preserving information for the target. (b) **KV Cache Optimization**: the target model's KV cache is shared with or reused by the drafter through learned projections or shared-backbone designs, eliminating redundant multimodal encoding. (c) **Target-Informed Transfer**: intermediate features or hidden representations from the target model are transferred to guide the drafter at inference time, tightening draft–target alignment without retraining. (d) **Drafter-Target Alignment**: the drafter is trained to align with the target through feature-level distillation ($\mathcal{L}_{\text{feat}}$) or logit-level knowledge distillation ($\mathcal{L}_{\text{KD}}$), embedding compatibility into the drafter's weights.

Semantic Visual Compression. The core idea is to replace dense raw visual tokens with a compact semantic surrogate before drafting. ViSpec (Kang et al., 2025) uses a Q-Former-style (Li et al., 2023) adaptor to compress visual tokens into a small set of queries plus a global vector. HiViS (Xie et al., 2025) eliminates

visual-prefill cost entirely, replacing visual tokens with fused semantic embeddings from the target model. SpecFLASH (Wang et al., 2025c) applies latent-aware compression using target sub-top-layer features.

Adaptive and Dynamic Compression. Here the compression ratio is input-dependent rather than fixed, so the drafter can spend computation only where it is most useful. SpecVLM (Huang et al., 2025) adaptively applies pooling, convolution, or pruning based on token redundancy, supporting multiple compression modes within a single framework. DREAM (Hu et al., 2025b) uses target intermediate features to guide visual token selection, retaining only informative tokens. MSD (Lin et al., 2025a) decouples text and vision processing, effectively bypassing visual tokens during drafting. EdgeSD (Huang et al., 2026) introduces a bandwidth-aware dynamic image token merging (ITM) method that progressively merges similar image tokens across transformer layers using cosine similarity on key vectors, reducing both computational cost and inter-server transmission latency in vision-decoding disaggregated architectures.

Attention-Based Token Pruning. These methods use attention signals from prefill or early layers to identify low-salience tokens that can be pruned. STD (Zhang et al., 2025) selects the top-K visual KV cache entries per layer and per head based on prefill-stage attention scores for VideoLMs. SpecVLM (Video) (Ji et al., 2025) applies training-free pruning that retains high-attention tokens via top-p selection while spatially sub-sampling low-attention regions to preserve geometric structure; the feedback mechanism that drives this pruning is detailed in §4.2.3. HIPPO (Lv et al., 2026) fuses three complementary scoring signals (global semantic relevance, temporal redundancy, and spatial redundancy) into a holistic score for video token selection. SpecPrune-VLA (Wang et al., 2025a) extends attention-based pruning to Vision–Language–Action models, fusing *global action history* with local attention signals to identify which visual tokens are expendable with respect to the current action decision, demonstrating that attention-guided compression applies beyond VLMs.

Visual Computation Elimination. Rather than compressing the visual stream, this strategy removes it from the drafter altogether. Sparrow (Zhang et al., 2026) eliminates the visual KV cache from the drafter entirely via text-anchored window attention (VATA), relying instead on text hidden states that already encode visual semantics distilled from the target model.

Similarity-Based Token Selection. Token retention is driven by cross-modal similarity dynamics rather than raw attention magnitude. ParallelVLM (Kong et al., 2026) introduces UV-Prune, which replaces attention-score-based token selection with vision–text cosine similarity variation measured across the target model's early layers. By tracking how each video token's cross-modal relevance evolves through the network, UV-Prune avoids the positional bias of attention-guided methods and is fully compatible with FlashAttention.

3.3.2 KV Cache Optimization

Multimodal KV caches are substantially larger than text-only caches due to visual and temporal token sequences, making KV management a critical bottleneck for speculative decoding efficiency.

Several complementary strategies address this challenge across modalities. AASD (Yang et al., 2025) reuses the target's KV cache directly through learned projections, compressing the multimodal KV before cross-attention to make the speculative module tractable. FastVLM (Bajpai & Hanawal, 2025) takes a different approach, sharing the backbone between drafter and verifier to maintain a single KV cache for both stages and eliminate redundant KV computation. SpecVLM (Huang et al., 2025) optimizes KV management at the prefill stage, reducing the memory footprint of visual token caching. STD (Zhang et al., 2025) exploits attention sparsity by selecting only the top-K KV entries, reducing I/O cost while sharing all parameters with the target model so no additional GPU memory is needed.

3.3.3 Target-Informed Transfer

These methods reduce draft–target mismatch by feeding the drafter with signals derived from the target model itself.

DREAM (Hu et al., 2025b) selects target intermediate layers using attention entropy and injects features via cross-attention at each draft step. SPARROW (Zhang et al., 2026) applies target-informed transfer at two stages: at inference time, hidden state reuse (HSR) feeds the drafter with the target's penultimate-layer text hidden state, which already encodes internalized visual semantics from the target model; at training time, intermediate-layer visual state bridging (IVSB) extracts visual hidden states from the target's interaction-active middle layers as supervision for the drafter, filtering out low-level visual noise. PARALLELVLM (Kong et al., 2026) transfers the target model's early-layer vision–text similarity signals to guide draft-side visual token pruning, providing alignment-aware token selection without runtime feature injection overhead.

Draft recycling, where rejected tokens are locally repaired and reused rather than discarded, is a feedback mechanism triggered by the verification stage and is detailed in §4.2.3.

3.3.4 Drafter-Target Alignment

Complementing the inference-time signal injection of Target-Informed Transfer, a parallel design pattern aligns the drafter with the target *during training*, embedding compatibility into the drafter's weights or architecture so that no runtime overhead is incurred.

Architectural Inheritance. The drafter's capacity is distilled from the target's architecture via weight sharing or knowledge transfer. SSD (Lin et al., 2025b) trains a compact audio language model via knowledge distillation from the full TTS target (CosyVoice-2), exploiting the strong acoustic conditioning in speech synthesis to maintain high acceptance rates with minimal drafter capacity. FASTVLM (Bajpai & Hanawal, 2025) bridges the gap between the shallow (n-layer) draft path and the full (L-layer) target through an imitation network trained to mimic the remaining $L - n$ layers, with all backbone parameters frozen.

Feature-Level Distillation. Training explicitly matches the drafter to the target at the logit or hidden-state level. MASSV (Ganesan et al., 2025) employs a two-stage training protocol: projector pretraining on paired image-text data followed by self-data distillation from the target VLM, transferring multimodal reasoning capabilities into a compact drafter. SPEC-LLAVA (Huo et al., 2025) applies online logit distillation during drafter training, aligning the drafter's output distribution with the target's at each token position. SPECVLM (Huang et al., 2025) similarly employs online logit distillation alongside its elastic visual compressor, ensuring distribution-level compatibility between drafter and target during decoding.

Representation Alignment. Auxiliary modules are trained so that compressed inputs remain compatible with the target's internal feature space. VISPEC (Kang et al., 2025) trains a Q-Former-style (Li et al., 2023) vision adaptor to produce compressed visual tokens and a global visual vector aligned with the target's visual processing. HIVIS (Xie et al., 2025) conditions the drafter on precomputed semantic embeddings exported from the target model, augmented with step-aware residual vectors that encode decoding-position-specific information. SPECFLASH (Wang et al., 2025c) co-trains the drafter with latent-aware compression using the target's sub-top-layer features, ensuring the compressed visual tokens remain semantically compatible with the target's expectations.

This training-time alignment pattern is distinct from, and often complementary to, inference-time Target-Informed Transfer. For example, SPARROW (Zhang et al., 2026) combines both: training-time intermediate-layer visual state bridging (IVSB) aligns the drafter's representations, while inference-time hidden state reuse (HSR) provides runtime target signals.

4 Verification and Acceptance Stage

In each decoding step, the drafted tokens are verified in parallel to ensure the outputs align with the target model. This process determines the number of tokens accepted per step, a key factor impacting the overall speedup. We organize verification into execution strategies (§4.1) and verification optimization (§4.2).

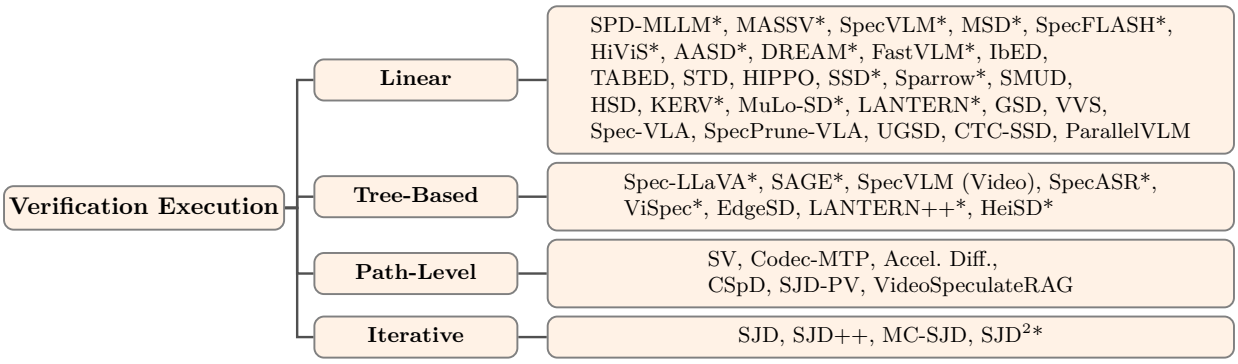

Figure 10: Sub-taxonomy of verification execution strategies (§4.1). Linear verification remains dominant; tree, path, and iterative approaches address domain-specific needs. * denotes methods requiring training/fine-tuning.

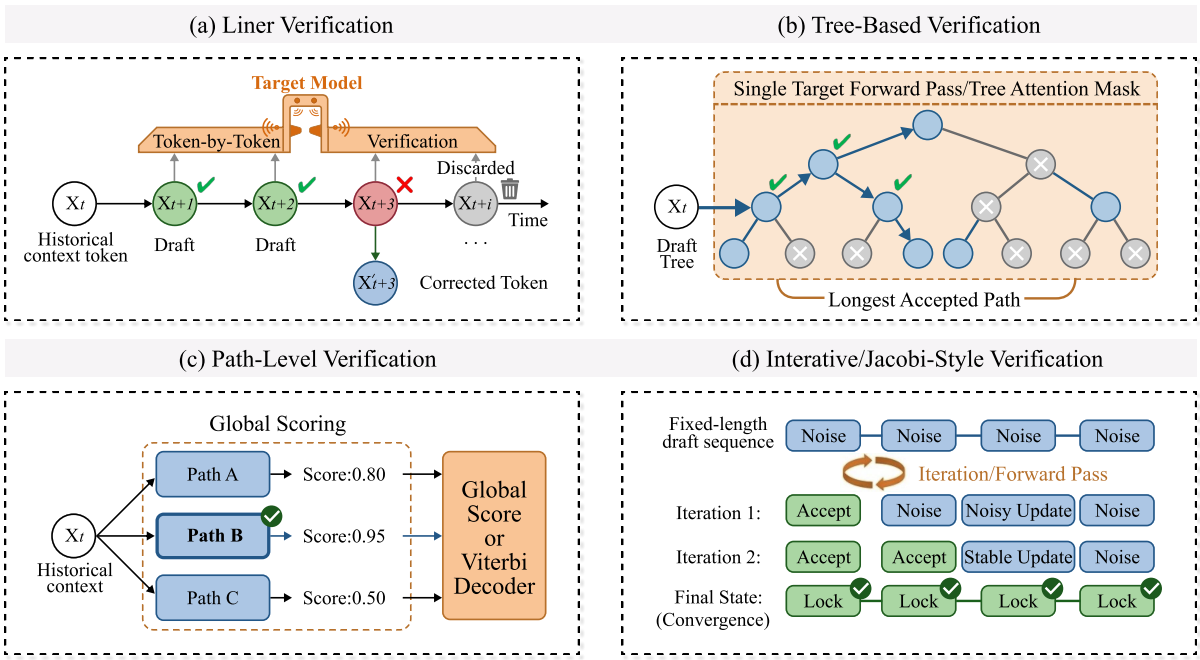

Figure 11: Illustration of four verification execution strategies in multimodal speculative decoding. (a) **Linear Verification**: draft tokens are verified sequentially left to right; the first rejected token is replaced by the target model's correction, and subsequent draft tokens are discarded. (b) **Tree-Based Verification**: multiple candidate continuations form a draft tree; the target model evaluates all branches in a single forward pass via tree attention masks and selects the longest accepted path. (c) **Path-Level Verification**: entire candidate paths are scored globally, and the highest-scoring path is selected rather than performing token-by-token accept/reject decisions. (d) **Iterative/Jacobi-Style Verification**: a fixed-length draft sequence initialized with noise is iteratively refined through forward passes; tokens that converge to stable predictions across consecutive iterations are progressively locked until the entire sequence reaches a fixed point.

4.1 Verification Execution

As shown in Figures 10 and 11, and Table 3, this section summarizes various verification execution strategies, encompassing linear verification (§4.1.1), tree-based verification (§4.1.2), path-level verification (§4.1.3), and iterative / Jacobi-style verification (§4.1.4).

Criteria	Formulation	Motivation
Distributional Match (§4.1.1)	$x_i \sim \text{Accept}\Big(\min\Big(1,\ \frac{p_{\text{target}}(x_i)}{p_{\text{draft}}(x_i)}\Big)\Big)$	Distribution-preserving decoding (lossless equivalence to target model)
Representation Consistency (§4.2.2)	$\mathcal{D}(\phi_{\text{target}}(x_i), \phi_{\text{draft}}(x_i)) < \tau$	Continuous-state validation (no discrete token identity)
Latent-Space Equivalence (§4.2.2)	$x_i^{\text{draft}} \in \mathcal{N}_\delta\Big(x_i^{\text{target}}\Big)$	Perceptual / codebook invariance (multiple tokens map to same meaning)

Table 3: Evolution of correctness criteria in speculative decoding. x_i denotes the i-th candidate token, $p_{\text{target/draft}}$ the target/draft model probability (Eq. 1), $\mathcal{D}$ a distance metric, $\phi_{\text{target/draft}}$ the representation function of the target/draft model, τ an acceptance threshold, and $\mathcal{N}_\delta(\cdot)$ a δ-neighborhood in the codebook embedding space. Classical text-only methods enforce distributional equivalence, while multimodal generation often adopts relaxed notions of consistency to accommodate continuous representations or perceptual invariances.

4.1.1 Linear Verification (Standard)

Standard verification evaluates K draft tokens left to right. The first token whose acceptance probability (Eq. 1) falls below a uniform random threshold terminates the draft; the target model's sample at that position replaces it, and drafting resumes. This procedure preserves the target distribution exactly.

Linear verification remains the most widely adopted baseline across multimodal speculative decoding.

In the VLM domain, the following methods all employ standard linear verification without modifying the verification algorithm itself: SPD-MLLM (Gagrani et al., 2024), MASSV (Ganesan et al., 2025), SpecVLM (Huang et al., 2025), MSD (Lin et al., 2025a), SpecFLASH (Wang et al., 2025c), HiViS (Xie et al., 2025), AASD (Yang et al., 2025), DREAM (Hu et al., 2025b), FastVLM (Bajpai & Hanawal, 2025), IbED (Lee et al., 2025), and TABED (Lee et al., 2026).

In the VideoLM domain, STD (Zhang et al., 2025), HIPPO (Lv et al., 2026), Sparrow (Zhang et al., 2026), and ParallelVLM (Kong et al., 2026) retain standard Leviathan-style accept/reject rules.

In the speech domain, SSD (Lin et al., 2025b) employs linear verification along a single draft sequence with speech-specific acceptance modifications. UGSD (Xue et al., 2026) also performs sequential left-to-right verification, combining it with a relaxed rank-based acceptance criterion and uncertainty-gated cloud offloading. CTC-SSD (Saon et al., 2026) employs linear verification of the CTC draft hypothesis through a single LLM forward pass, accepting the hypothesis if all token likelihoods exceed a threshold; otherwise it falls back to AR decoding from the longest accepted prefix. SMUD (Okabe & Yamamoto, 2025) dynamically verifies candidates across two parallel decoding hypotheses ("still inside mask" vs. "exited mask"), selecting the winning path based on a mixed CTC-AR probability score.

In the T2I (Vision AR) domain, LANTERN (Jang et al., 2025a), GSD (So et al., 2025b), and VVS (Dong et al., 2025) employ sequential left-to-right verification for visual token generation, though LANTERN and GSD relax the acceptance criterion via latent-space neighborhood or grouped acceptance, and VVS dynamically skips verification for high-confidence tokens.

While the execution strategy in all these methods is linear, many introduce domain-specific acceptance relaxations or cost optimizations, which are discussed later in this section.

4.1.2 Tree-Based Verification

Tree-based verification evaluates multiple candidate continuations simultaneously using structured attention masks, forming a token tree. The target model processes the entire tree in parallel and selects the longest accepted path. Tree-based verification increases per-step cost compared to linear verification but yields more accepted tokens per step, providing net speedups when per-token draft accuracy is moderate.

In the VLM domain, Spec-LLaVA (Huo et al., 2025) is the primary method employing strict tree-based verification. A token-tree attention mask enables the target to evaluate all branches in a single forward pass. Verification proceeds leaf-to-root: deeper paths are tried first (since they yield more accepted tokens on success), and a mismatch at any depth truncates the speculative block at the failure point. ViSpec (Kang et al., 2025) similarly uses a tree-based speculative mechanism during generation. SAGE (Tong et al., 2026) extends this paradigm with entropy-guided shaping that dynamically determines the tree topology the target verifies. EdgeSD (Huang et al., 2026) employs tree-based verification via masked tree attention on the cloud-side target VLM, verifying the adaptive token tree generated by the edge-side drafter in a single forward pass and selecting the longest accepted branch.

In the VLA domain, HeiSD (Zheng et al., 2026b) adapts tree-based verification by constructing a sequence-wise tree from top-K retrieved drafts, where each node represents a kinematically correlated token group (position, rotation, gripper) rather than a single token. Verification proceeds via depth-first search over chains, combining sequence-wise relaxed acceptance with adaptive verify-skip to select the longest acceptable action sequence.

In the VideoLM domain, SpecVLM (Video) (Ji et al., 2025) adopts EAGLE-style (Li et al., 2024a) static tree structures with tree attention masks, verifying multi-branch candidate trees generated from pruned video tokens.

In the speech domain, SpecASR (Wei et al., 2025) employs a two-pass sparse tree structure that branches at positions of high uncertainty and dynamically constructs a tree, moving beyond pure linear sequences to multi-sequence branching for the target to evaluate.

In the T2I (Vision AR) domain, LANTERN++ (Jang et al., 2025b) extends tree-based verification to continuous domains: the target evaluates all branches of a static speculation tree and selects the longest accepted path, where acceptance is defined over codebook embedding neighborhoods rather than exact token matches.

4.1.3 Path-Level Verification

Path-level verification is a coarse-grained consistency check that operates over entire candidate trajectories rather than individual tokens. Several recent methods do not introduce an explicit verifier, but their selection rules play an analogous role by validating global hypotheses against the target model or process.

In the VLM domain, SV (Speculative Verdict) (Liu et al., 2025b) performs trajectory-level validation for reasoning tasks. The verdict model does not perform token-by-token accept/reject decisions. Instead, it receives multiple complete reasoning paths as evidence and synthesizes a new final answer in a single inference call. A consensus filter based on cross-model NLL scores ("how likely would other models find this answer reasonable?") pre-screens paths before presenting them to the verdict model, reducing its input cost.

In the VideoLM domain, VideoSpeculateRAG (Li & Liu, 2026) applies candidate-level verification to video RAG, verifying full answer candidates rather than token prefixes, effectively treating each retrieved document as an independent speculative branch. Each retrieved document produces an independent candidate answer via a lightweight draft VLM; the verifier then performs two-stage candidate reranking combining tolerance-based reliability filtering with entity-alignment scoring.

In the speech domain, Codec-MTP (Nguyen et al., 2025) performs sequence-level selection. Rather than accepting or rejecting individual tokens, Codec-MTP applies HMM/Viterbi global optimal path selection over multiple candidate codec token sequences. This sequence-level verification selects the most likely path given the full generative model, analogous to diffusion's trajectory-level verification but operating over discrete codec states. A top-k reduction of the state space makes Viterbi path scoring computationally feasible.

In the T2I (Vision AR) domain, CSpD (Wang et al., 2024) enforces density-level consistency for continuous-valued visual AR models through acceptance-rejection sampling over the draft–target density ratio. SJD-PV (Yu et al., 2026) (Speculative Jacobi Decoding with Phrase Verification) elevates verification granularity from individual tokens to the *phrase level*. Observing that visual semantics are encoded across contiguous token sequences, SJD-PV constructs a *phrase library* via BPE-style iterative merging on large-scale datasets to extract recurring semantic priors. During verification, if a draft sequence matches a library entry, a joint acceptance score $\log R_{\mathrm{p}} = \sum_k (\log p(v_k) - \log q(v_k))$ validates the phrase as a single unit, where R_{p} is the phrase-level acceptance ratio, v_k is the k-th token in the matched phrase, and $p(\cdot)$, $q(\cdot)$ are the target and draft token probabilities, respectively. This joint criterion is more efficient than token-wise verification because aggregation prevents individual low-confidence tokens from prematurely truncating a high-confidence block. As a training-free module, SJD-PV augments existing SJD variants.

In the diffusion domain, path-level verification evaluates whether a proposed denoising trajectory segment constitutes a valid draw from the target diffusion process, requiring coupling-based acceptance criteria (De Bortoli et al., 2025).

4.1.4 Iterative / Jacobi-Style Verification

Iterative verification evaluates whether the Jacobi fixed-point iteration has converged, rather than comparing draft tokens against a separate target model's output. This paradigm is unique to self-drafting methods that use the target model itself in Jacobi mode.

The SJD family (Teng et al., 2025b;a; So et al., 2025a; Teng et al., 2025c) defines probabilistic stability criteria: tokens whose predictions remain stable across consecutive iterations are accepted simultaneously, bypassing the standard draft–target verification framework entirely. MC-SJD (So et al., 2025a) strengthens convergence detection by applying maximal coupling between iterations, maximizing the probability that consecutive steps sample identical tokens. SJD^2 (Teng et al., 2025c) refines unaccepted tokens along a structured denoising trajectory rather than re-sampling independently, yielding smoother convergence and higher per-step acceptance rates.

4.2 Verification Optimization

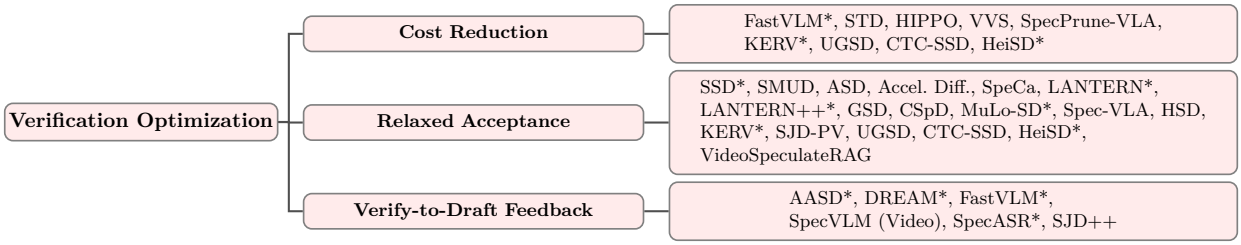

Figure 12: Sub-taxonomy of verification optimization strategies (§4.2). These strategies recur across modalities. * denotes methods requiring training/fine-tuning.

As with draft generation, several verification-side optimization strategies recur across modalities, as shown in Figures 12 and 13.

4.2.1 Cost Reduction

The target model's verification forward pass typically dominates the speculative decoding pipeline's computational cost; reducing its per-step cost is therefore critical for achieving net latency gains.

Confidence-Based Verification Skipping. The shared idea is to bypass expensive target verification when the draft already appears reliable enough. VVS (Dong et al., 2025) implements this at token granularity, building on two observations: *verification redundancy* (many draft tokens would be accepted regardless) and *stale feature reusability* (cached target features remain informative across consecutive steps). This

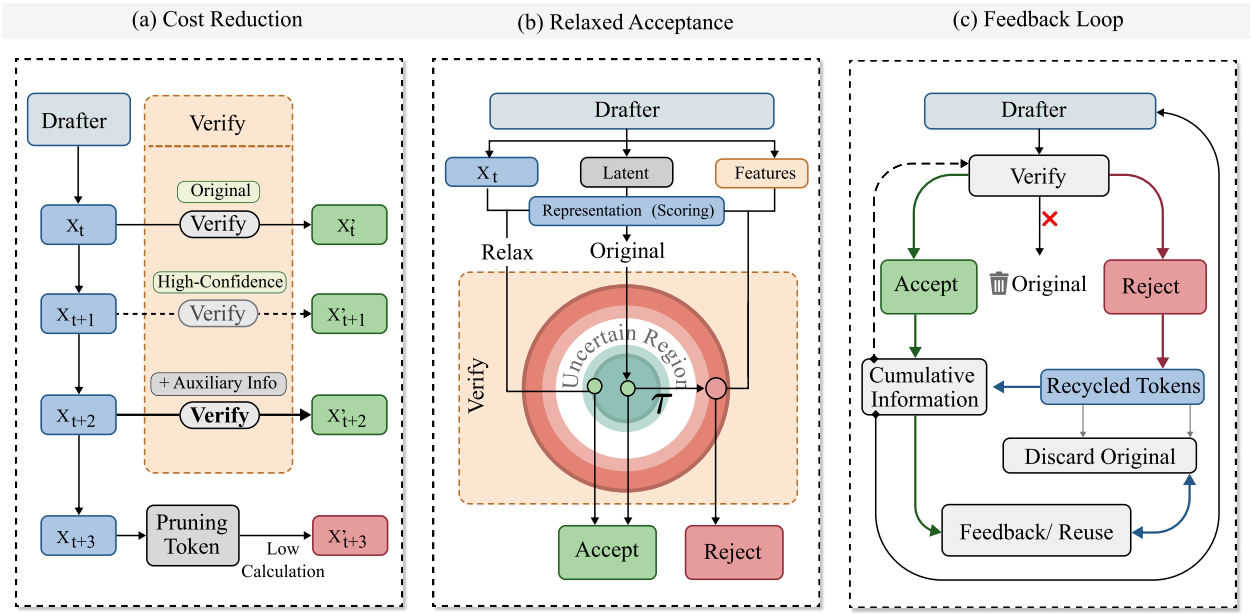

Figure 13: Illustration of three verification optimization strategies in multimodal speculative decoding. (a) **Cost Reduction**: verification overhead is reduced by selectively skipping verification for high-confidence tokens or leveraging auxiliary information, avoiding redundant target model forward passes. (b) **Relaxed Acceptance**: the strict distributional match criterion is replaced with representation-level or latent-space proximity checks; draft outputs—whether discrete tokens, continuous latent states, or intermediate features—are accepted when they fall within an acceptable region around the target's predictions, accommodating perceptual or functional equivalence across modalities. (c) **Feedback Loop**: verification outcomes are fed back to improve subsequent drafting; accepted tokens accumulate context while rejected tokens are recycled or reused rather than discarded, creating a closed-loop mechanism that progressively improves draft quality.

verification skipping parallels SpeCa (Liu et al., 2025a)'s forecast gating, where accurate cached predictions bypass full recomputation, but operates at the token level rather than the feature level. UGSD (Xue et al., 2026) takes a complementary approach: rather than skipping verification for confident tokens, it skips *cloud offloading* entirely; low-entropy token blocks remain on the edge device without invoking the cloud verifier, reducing communication and computation overhead while reserving cloud capacity for the most uncertain predictions. CTC-SSD (Saon et al., 2026) implements an analogous entropy-based gating at the CTC level: if all frame-level entropies of the CTC output fall below a threshold τ_{CTC}, the greedy CTC hypothesis is accepted as final *without any LLM verification pass*, entirely bypassing the most expensive stage of the pipeline. HeiSD (Zheng et al., 2026b) extends verification skipping from token-level to trajectory-segment level: when feature similarity between retrieved drafts and historical trajectories exceeds a learned threshold, the entire verification pass is bypassed, with the threshold adapting online via task-completion feedback. While the preceding methods skip verification proactively, KERV (Zheng et al., 2026a) avoids target recomputation reactively: upon rejection, a Kalman Filter predicts the remaining action trajectory from kinematic history, entirely bypassing GPU-side re-drafting.

Cross-Stage Computation Reuse. Verification becomes cheaper here because computation produced during drafting is reused instead of recomputed. FastVLM (Bajpai & Hanawal, 2025) shares the KV cache between draft and verification stages within a single backbone: the first n layers' KV entries, computed during drafting, transfer directly to full-model verification, eliminating redundant multimodal encoding. STD (Zhang et al., 2025) similarly reassigns computational responsibility: the target model $\mathcal{M}$ (with full KV cache) handles verification exclusively, while the sparse surrogate $\mathcal{M}_s$ (with reduced KV) handles drafting, sharing all parameters so no additional model memory is required. HIPPO (Lv et al., 2026) takes a complementary

approach: rather than eliminating redundant KV computation, it hides verification latency by pipelining target verification of batch t with draft generation of batch $t+1$ via double-buffer KV cache management.

Input Pruning for Shared-Backbone Verification. The savings here come from shrinking the token set that the shared backbone must process in both stages. SpecPrune-VLA (Wang et al., 2025a) achieves this through its action-aware pruning framework: visual token count entering the shared backbone is reduced, making both draft and verification forward passes cheaper. Its self-speculative design further avoids maintaining a separate draft model.

4.2.2 Relaxed Acceptance

Standard speculative decoding enforces exact distribution matching. Multiple modalities, including VLA, speech, T2I (Vision AR), and diffusion, benefit from relaxing this requirement when perceptual or functional equivalence suffices. For diffusion models, verification is naturally defined as the consistency of the proposed trajectory under the target dynamics, rather than token-wise acceptance.

Perceptual and Functional Tolerance. Exact token agreement is replaced by modality-specific notions of perceptual or functional equivalence. In the VLA domain, Spec-VLA (Wang et al., 2025b) accepts draft action tokens whose continuous control signals fall within a task-dependent distance tolerance of the target action. KERV (Zheng et al., 2026a) deepens this paradigm by replacing Spec-VLA's static acceptance threshold with a *kinematic-based dynamic adjustment strategy* that tunes the tolerance based on real-time kinematic variability; its Kalman Filter fallback mechanism for rejected tokens is discussed earlier in the drafting section. HeiSD (Zheng et al., 2026b) introduces *sequence-wise* relaxed acceptance: rather than evaluating tokens individually, it groups kinematically correlated action dimensions (position, rotation, gripper) into sequences and accepts an entire sequence when its aggregate bias bias_{seq} (computed over the full sequence) remains within tolerance, even if the bias of an individual action dimension a_j, denoted bias_{a_j}, is larger. In the VideoLM domain, VideoSpeculateRAG (Li & Liu, 2026) applies *tolerance-based candidate selection*: rather than requiring exact token matching, it retains all candidates whose reliability scores fall within a margin δ of the maximum, then reranks this tolerant set by entity-alignment similarity to select the final answer. In the VLM domain, HSD (Liao et al., 2026) introduces τ-matching for document parsing: a drafted region-level Markdown sequence is accepted if its edit distance to the target falls below a formatting tolerance threshold τ, permitting minor whitespace and markup variations that are semantically equivalent. In the speech domain, SSD (Lin et al., 2025b) relaxes the standard acceptance criterion by introducing a perceptual tolerance parameter β: acoustically equivalent tokens, i.e., those producing perceptually indistinguishable audio despite differing in discrete codec representation, are accepted even when exact distribution matching fails. UGSD (Xue et al., 2026) adopts a rank-based acceptance rule for speech emotion captioning: a drafted token is accepted if it falls within the top-R most probable tokens under the cloud verifier's distribution, replacing strict exact matching with a practical relaxation suited to open-ended caption generation. CTC-SSD (Saon et al., 2026) relaxes acceptance differently: the greedy CTC hypothesis is accepted if all token likelihoods under the LLM distribution exceed a threshold τ_{SLM}, replacing exact token matching with a plausibility check that permits the verifier to endorse acoustically grounded but lexically distinct hypotheses. Across all these modalities, the common insight is that multiple distinct tokens can function equivalently when evaluated through a modality-appropriate perceptual or task-specific metric.

Latent-Space and Continuous-Density Acceptance. Acceptance is defined by proximity in a latent or continuous space rather than exact token identity. In visual generation, exact token matching yields impractically low acceptance rates due to codebook redundancy. LANTERN (Jang et al., 2025a) and LANTERN++ (Jang et al., 2025b) relax exact token matching to latent-space neighborhood acceptance: a draft token is accepted if it falls within a distance threshold of the target's prediction in the codebook embedding space. GSD (So et al., 2025b) adopts grouped acceptance, treating visually equivalent token clusters as interchangeable. CSpD (Wang et al., 2024) extends relaxation to continuous-valued token spaces through density-ratio acceptance-rejection sampling, enabling theoretically grounded verification in continuous output spaces. Mulo-SD (Peruzzo et al., 2026) introduces *local spatial relaxation*: rather than rejecting all tokens after the first failure in raster-scan order, it accepts each draft token independently when the pooled

probability over its k-nearest latent neighbors exceeds a threshold τ, and resamples only within a spatial neighborhood of radius l (in discrete token positions) around rejected positions, exploiting the locality of visual AR attention patterns.

Coupling-Based and Exchangeability-Based Acceptance. Here correctness is established through structural properties of the diffusion process itself rather than discrete token matching. Accelerated Diffusion Sampling (De Bortoli et al., 2025) verifies proposals through reflection maximal coupling: the proposed future sample is accepted if the coupled SDE trajectory remains within a valid region defined by the score function geometry, providing a theoretically grounded acceptance criterion that preserves the target diffusion distribution. ASD (Hu et al., 2025a) provides theoretical guarantees through the exchangeability property of stochastic localization: because permuting denoising timestep orderings does not change the output distribution, self-proposed multi-step jumps are provably correct without requiring explicit verification.

Forecast Gating. The verifier selectively trusts cheap feature forecasts and falls back to full recomputation only when the forecast degrades. SpeCa (Liu et al., 2025a) applies a gating mechanism to Taylor-forecasted features: if the discrepancy between the cached forecast and the actual model computation exceeds a threshold, the forecast is rejected and the full model recomputes that step.

Dual-Hypothesis Boundary Detection. The key idea is to keep two competing boundary hypotheses alive until the verifier can resolve the masked-region ambiguity. Unlike traditional speculative decoding where the verifier merely checks if a proposed token is correct, SMUD (Okabe & Yamamoto, 2025) tackles the fundamental ambiguity of masked decoding: when the AR decoder processes a token, it does not know if the token belongs *inside* the mask region or has *exited* into the known post-mask text. SMUD (Okabe & Yamamoto, 2025) solves this by simultaneously maintaining two parallel decoding hypotheses: H_{in} (assuming decoding continues inside the mask) and H_{out} (assuming the mask has ended). The verifier selects the correct trajectory by comparing a joint CTC-decoder score, effectively using the autoregressive decoder as a soft verifier to probabilistically determine the true mask boundary while preserving CTC prefix scores.

4.2.3 Verify-to-Draft Feedback

Rather than treating verification as a one-directional judgment, several methods create feedback loops where verification-side information actively improves subsequent draft quality.

Training-Time Alignment. Feedback is injected during training by baking draft–target compatibility into the drafter's parameters. The target-informed transfer mechanisms described in the drafting section, including AASD (Yang et al., 2025)'s T-D Attention and DREAM (Hu et al., 2025b)'s entropy-adaptive feature injection, function as implicit feedback loops: by aligning draft representations to target representations during training, these methods ensure that the verification step encounters fewer out-of-distribution tokens at inference time, increasing the effective acceptance rate α without adding inference-time overhead. FastVLM (Bajpai & Hanawal, 2025) creates a more explicit feedback loop: rejected tokens from verification are corrected by the full model and fed back to improve the imitation network through iterative imitation learning, progressively tightening draft–target alignment over decoding rounds.

Verifier Attention as Pruning Signal. Verifier attention is reused as an explicit signal for the drafter's next round of token selection. SpecVLM (Video) (Ji et al., 2025) applies this principle to video token pruning. After each verification step, the verifier's attention distribution over video tokens determines which tokens the drafter retains in subsequent rounds, creating an adaptive pruning loop where verification-side information continuously refines draft-side input selection.

Draft Recycling. Rejected draft content is treated as partial progress to be repaired or reused, rather than thrown away. SpecASR (Wei et al., 2025) (speech) transforms verification from "reject and restart" to "repair and reuse": rejected draft tokens are locally recycled, merged, reused, or expanded at uncertain positions, converting wasted draft computation into usable partial results. SJD++ (Teng et al., 2025a) (T2I, Vision AR) achieves analogous recycling through *high-confidence token reuse*: within Jacobi iterations,

tokens whose predictions remained stable across two consecutive steps are locked rather than re-sampled, converting otherwise-discarded iteration results into convergence acceleration. Both methods share the insight that partially correct draft information carries value beyond binary accept/reject decisions; the reject signal from verification is a structured feedback that guides subsequent drafting rather than a mere termination condition.

5 Frameworks and Systems

As speculative decoding for multimodal models matures, it becomes essential to examine the production inference frameworks that support it. Table 4 summarizes the major frameworks, their speculative decoding capabilities, and their current level of multimodal support.

vLLM. vLLM (Kwon et al., 2023) is a high-throughput LLM serving engine built on PagedAttention for efficient KV cache management and continuous batching. The engine provides the most comprehensive speculative decoding support among production frameworks, implementing model-based methods (EAGLE (Li et al., 2024a), EAGLE-3 (Li et al., 2025b), MTP, and draft models) as well as simpler n-gram and suffix decoding. vLLM's speculative decoding is algorithmically validated to be lossless, maintaining the same output distribution as standard decoding. As of v0.12.0, vLLM has begun integrating multimodal-aware speculative decoding: the Qwen3-VL model class natively supports both EAGLE and EAGLE-3 speculation (PR #29594), and broader multimodal draft model support is under active development (Issue #33458). However, deeper multimodal-specific optimizations such as visual token compression or heterogeneous KV cache management have not yet been incorporated into the speculative decoding pipeline.

SGLang. SGLang (Zheng et al., 2024) is an inference engine optimized for structured generation and multi-turn conversations through RadixAttention, which efficiently reuses shared KV cache prefixes. It supports speculative decoding via EAGLE-2 (Li et al., 2024b)/EAGLE-3, MTP, a standalone draft model mode, and an n-gram variant, and has been actively expanding support for multimodal models, including Vision–Language Models. Its development roadmap for 2025 explicitly prioritized speculative decoding optimizations, including adaptive batch-size-aware speculation. However, like vLLM, SGLang's speculative decoding currently targets text-only token prediction without multimodal-specific adaptations.

TensorRT. NVIDIA provides two inference frameworks under the TensorRT brand that support speculative decoding, targeting datacenter and edge deployment scenarios, respectively.

TensorRT-LLM. TensorRT-LLM (NVIDIA, 2024) is NVIDIA's datacenter-oriented inference optimization framework built on a dual-backend architecture that combines a TensorRT engine path with a PyTorch-native path, applying mixed-precision quantization (FP8, FP4, INT4 AWQ, INT8 SmoothQuant), layer fusion, and kernel auto-tuning for GPU-optimized deployment. It offers the broadest speculative decoding method coverage among NVIDIA's frameworks, supporting draft-target model pairs, EAGLE (1/2/3), Medusa (Cai et al., 2024), ReDrafter, Multi-Token Prediction (MTP), lookahead decoding, and n-gram methods. While TensorRT-LLM supports multimodal model serving (Qwen2-VL, LLaVA-NeXT, Llama 3.2 Vision, among others) and speculative decoding independently, its speculative decoding pipeline is primarily designed for text-only LLMs and does not yet provide end-to-end multimodal-aware speculation.

TensorRT Edge-LLM. TensorRT Edge-LLM (NVIDIA, 2026) is a separate lightweight C++ inference runtime purpose-built for embedded platforms such as NVIDIA DRIVE AGX Thor and Jetson Thor. Unlike TensorRT-LLM, it focuses on minimal dependencies and low resource footprint for real-time edge applications. Its speculative decoding support is limited to EAGLE-3, but notably it provides end-to-end multimodal-aware speculative decoding: it supports multi-batch EAGLE-3 speculation for both LLMs and VLMs, with native support for Qwen2/2.5/3-VL, InternVL3, and Phi-4-Multimodal. Industry partners including Bosch, ThunderSoft, and MediaTek have adopted TensorRT Edge-LLM for in-vehicle AI assistants and cabin monitoring, making it one of the few production frameworks where multimodal speculative decoding is deployed in real-world applications.

LMDeploy. LMDeploy (MMRazor and MMDeploy Teams, 2023) provides high-performance inference through its TurboMind C++ backend and PyTorch backend, featuring continuous batching and efficient CUDA kernels. It implements speculative decoding through EAGLE-3 with a separate draft model and DeepSeek-style Multi-Token Prediction (MTP), both exposed via a unified `SpeculativeConfig` interface on the PyTorch backend. Although LMDeploy supports multimodal model deployment (VLMs), its speculative decoding capabilities remain text-focused and are still marked as experimental.

Text Generation Inference (TGI). TGI (Hugging Face, 2023) is Hugging Face's production serving framework, built with a Rust/Python/gRPC stack and used to power Hugging Chat and the Hugging Face Inference API/Endpoints. Its speculative decoding support is intentionally narrow: only Medusa-style multi-head speculation (with tree attention) and n-gram prompt lookup are offered. Notably, TGI was the first major serving framework to ship Medusa in a production setting. TGI provides broad VLM serving coverage (Idefics 2/3, LLaVA-NeXT, Qwen2-VL, Qwen2.5-VL, PaliGemma, Gemma 3, among others), but its speculative decoding pipeline does not currently extend multimodal-aware speculation to these models.

Hugging Face Transformers. The Hugging Face Transformers library (Wolf et al., 2019) provides a widely adopted assisted decoding API (exposed through `generate()`) that implements several forms of speculative decoding: a separate draft model via the `assistant_model` argument, prompt-lookup (n-gram) speculation via `prompt_lookup_num_tokens`, self-speculative early-exit via `assistant_early_exit`, and Universal Assisted Decoding (UAD) for cross-tokenizer speculation. Dynamic speculation lookahead is the default operation mode since v4.45.0. Among the surveyed methods, HIPPO (Lv et al., 2026) explicitly builds on Transformers v4.57.0, and DREAM (Hu et al., 2025b) uses "official Hugging Face implementations" as its model backends. More broadly, methods targeting LLaVA-series, Qwen-VL-series, and InternVL-series models inherit the Hugging Face interfaces, making it the de facto prototyping platform for multimodal speculative decoding research.

Speculative Decoding Algorithm Libraries and Ecosystem Tools. Beyond production serving frameworks, several open-source libraries provide reusable implementations of speculative decoding techniques. EAGLE and its successor EAGLE-2 offer feature-level auto-regression heads with tree-structured verification, directly adopted by LANTERN (Jang et al., 2025a), SpecVLM (Huang et al., 2025), SAGE (Tong et al., 2026), and DREAM (Hu et al., 2025b). Medusa provides multi-head parallel drafting, used as a baseline by DREAM (Hu et al., 2025b). Spec-Bench (Xia et al., 2024) offers standardized evaluation protocols (speedup ratio, acceptance length, temperature sensitivity) that multimodal methods widely adopt for reporting results, despite being focused on text-only scenarios. Two emerging ecosystem tools further bridge research and deployment: SpecForge (Li et al., 2025a) is a training framework for EAGLE-3 draft heads natively integrated with SGLang, whose roadmap explicitly includes VLM support; and Speculators (Red Hat, 2025) defines a standardized Hugging Face model format for speculator checkpoints with direct vLLM integration, reducing the friction of porting research drafters into production.

Discussion. A narrowing but still persistent gap remains between speculative decoding and multimodal model serving in production inference frameworks. While all major frameworks now support both capabilities independently, only a few have begun combining them: TensorRT Edge-LLM provides end-to-end multimodal EAGLE-3 speculation for edge deployment, and vLLM has introduced EAGLE/EAGLE-3 support for Qwen3-VL as of v0.12.0, with broader multimodal draft model support under active development. However, these initial integrations remain limited in scope, typically restricted to specific model families and a single speculation method, without incorporating deeper multimodal-specific optimizations such as visual token compression, heterogeneous KV cache management, or relaxed verification criteria tailored to vision tokens. The surveyed research methods universally implement their multimodal SD innovations as standalone codebases (typically built on Hugging Face Transformers or EAGLE), rather than as extensions to production serving systems. Bridging this gap fully, i.e., integrating the full spectrum of multimodal-aware speculation techniques into frameworks like vLLM and SGLang with broad model coverage and production-grade robustness, remains a critical engineering direction for enabling practical deployment (§7).

A complementary trend is emerging at the orchestration layer above single-engine frameworks: NVIDIA Dynamo (NVIDIA, 2025) is an open-source distributed inference serving framework that uses vLLM, TensorRT-

Table 4: Production inference frameworks with speculative decoding support. "SD Methods" lists the supported speculative decoding algorithms. "MM" indicates native multimodal model support. "MM SD" indicates whether the speculative decoding pipeline can operate in a multimodal-aware manner rather than falling back to text-only speculation. △ denotes partial support limited to specific model families or methods.

Framework	SD Methods	MM	MM SD
vLLM (Kwon et al., 2023)	EAGLE, EAGLE-3, MTP, Draft Model, n-gram, Suffix	✓	△[a]
SGLang (Zheng et al., 2024)	EAGLE-2, EAGLE-3, MTP, Standalone Draft, n-gram	✓	×
TensorRT-LLM (NVIDIA, 2024)	EAGLE (1/2/3), Medusa, ReDrafter, MTP, Lookahead, Draft Model, n-gram	✓	×
TensorRT Edge-LLM (NVIDIA, 2026)	EAGLE-3	✓	✓[b]
LMDeploy (MMRazor and MMDeploy Teams, 2023)	EAGLE-3, DeepSeek MTP	✓	×
TGI (Hugging Face, 2023)	Medusa (TreeMask), n-gram	✓	×
HF Transformers (Wolf et al., 2019)	Draft Model, Prompt Lookup (n-gram), Self-Speculative (Early-Exit), UAD	✓	×[c]

[a]As of v0.12.0, EAGLE/EAGLE-3 multimodal SD is supported for Qwen3-VL (PR #29594); broader multimodal draft model support is under development (Issue #33458). [b]Supports multi-batch EAGLE-3 for VLMs including Qwen2/2.5/3-VL, InternVL3, and Phi-4-Multimodal on embedded platforms. [c]Serves as the de facto prototyping backend for multimodal SD research methods, though its assisted decoding API itself lacks multimodal-specific optimizations.

LLM, or SGLang as backend engines and coordinates disaggregated prefill/decode execution across multiple GPUs. Dynamo supports disaggregated EAGLE-3 speculative decoding across prefill and decode workers, and has recently introduced multimodal encode/prefill/decode (E/P/D) disaggregation with an embedding cache, reporting substantial TTFT improvements on image workloads. This orchestration-layer approach echoes the vision-decoding disaggregation architecture of EdgeSD (Huang et al., 2026) in the edge-cloud setting and suggests that decoupling multimodal encoding from LLM inference is emerging as a production pattern for multimodal serving at scale.

6 Comparison and Benchmarking

This section provides a systematic comparison of representative multimodal speculative decoding methods. We first present a cross-modal comparative summary (§6.1), then discuss the first standardized VLM benchmark (§6.2), and finally address benchmarking for other modalities (§6.3).

6.1 Cross-Modal Comparative Summary

Table 5 highlights four fundamental differences from text-only speculative decoding: (1) **Drafting architecture is modality-dependent**: independent drafters dominate VLMs, while other modalities exhibit a broader mix of shared-backbone, Jacobi self-drafting, and drafter-free approaches; (2) **T2I generation reveals two distinct paradigms**: relaxed-acceptance dual-model methods versus Jacobi self-drafting; (3) **Tuning-free deployment remains a primary goal**: yet the highest reported speedups still generally require dedicated drafter training; (4) **Verification criteria diverge by output space**: VLMs and Video–Language Models largely retain exact-match acceptance, while modalities with continuous, latent, perceptual, or task-tolerant outputs more often require relaxed or convergence-based criteria.

First, *Independent Drafting* dominates Vision–Language and Video–Language Models, mirroring text-only practice. However, multimodal drafters increasingly incorporate target-model features to improve speculation accuracy (DREAM (Hu et al., 2025b), AASD (Yang et al., 2025)). Conversely, domains with strong self-drafting structures, such as video KV-splits (STD (Zhang et al., 2025)), layer-sharing (FastVLM (Bajpai & Hanawal, 2025)), and Jacobi iteration (the SJD (Teng et al., 2025b) family), adopt *Shared Backbone*

Methods	Drafting			Verification		Representative Target	Speedup (rep.)
	Draft Mechanism	Architecture / Approach	Tuning-free	Criterion	Pattern		
Vision–Language Models							
SPD-MLLM	Independent	Text-Only LM	× (Training)	Strict	Linear	LLaVA (LLaMA)	$\leq 2.37\times$
MASSV	Independent	Small VLM	× (Distillation)	Strict	Linear	Qwen2.5-VL, Gemma 3	$\leq 1.46\times$
ViSpec	Independent	Visual Adaptor	× (Tuning)	Strict	Tree	LLaVA-1.6, Qwen2.5-VL	$\leq 3.22\times$
SpecVLM	Independent	Visual Compressor	× (Tuning)	Strict	Linear	LLaVA, Qwen2-VL	$\leq 2.9\times$
FastVLM	Shared Backbone	Early Exiting	× (Distillation)	Strict	Linear	LLaVA-1.5	$1.55\times \sim 1.85\times$
MSD	Independent	Text-Vision Decouple	✓	Strict	Linear	LLaVA-1.5	$\leq 2.46\times$
SpecFLASH	Independent	Semi-AR Heads	× (Tuning)	Strict	Linear	LLaVA, Qwen-VL	$\leq 2.55\times$
Spec-LLaVA	Independent	Token Tree	× (Tuning)	Strict	Tree	LLaVA-1.5	$\leq 3.28\times$
SAGE	Independent	Adaptive Tree	× (Tuning)	Strict	Tree	LLaVA-OV, Qwen2.5-VL	$\leq 3.36\times$
HiViS	Independent	Visual Token Hiding	× (Tuning)	Strict	Linear	Qwen2.5-VL	$\leq 3.15\times$
AASD	Independent	T-D Attention	× (Tuning)	Strict	Linear	LLaVA	$\leq 2.0\times$
DREAM	Independent	Target-Informed	× (Tuning)	Strict	Linear	LLaVA-1.6	$\leq 3.6\times$
IbED	Independent	Multi-Prompt Ensemble	✓	Strict	Linear	LLaMA, LLaVA-1.5	$1.06\times \sim 1.23\times^{\dagger}$
TABED	Independent	Test-Time Adaptive Weighting	✓	Strict	Linear	LLaVA-1.5, LLaVA-NeXT	$\leq 1.74\times$
EdgeSD	Independent	Edge-Cloud Decoupling	✓	Strict	Tree	LLaVA-OV, InternVL2.5	$3.04\times \sim 5.12\times$
SV (Verdict)*	Independent	Multi-VLM Consensus	✓	Relaxed	Path	Qwen2.5-VL	N/A
HSD‡	Independent	Pipeline Draft	✓	Relaxed	Linear	Qwen3-VL	$\leq 4.89\times$
Vision–Language–Action Models							
Spec-VLA	Independent	Small VLA	× (Training)	Relaxed	Linear	OpenVLA	$\approx 1.42\times$
SpecPrune-VLA	Shared Backbone	Action-Aware Pruning	✓	Strict	Linear	OpenVLA-OFT	$1.46\times \sim 1.57\times$
KERV	Independent	Kalman-Rectified Draft	× (Training)	Relaxed	Linear	OpenVLA (LIBERO)	$1.48\times \sim 1.57\times$
HeiSD	Independent	Hybrid Retrieval + Drafter	× (Training)	Relaxed	Tree	OpenVLA (LIBERO)	$\leq 2.45\times$
Video–Language Models							
STD	Shared Backbone	Sparse KV Routing	✓	Strict	Linear	Qwen2-VL, LLaVA-OV	$\leq 1.94\times$
SpecVLM (Video)	Independent	Token Tree	✓	Strict	Tree	LLaVA-OV	$\leq 2.68\times$
HIPPO	Shared Backbone	Pipeline Overlap	✓	Strict	Linear	LLaVA-OV	$\leq 3.51\times$
Sparrow	Independent	Hidden State Reuse	× (Training)	Strict	Linear	LLaVA-OV, Qwen2.5-VL	$\leq 2.82\times$
VideoSpeculateRAG	Independent	Per-Doc Parallel Draft	✓	Relaxed	Path	Qwen2.5-VL	$\approx 2\times$
ParallelVLM	Independent	Verifier Pruning + Pipeline	✓	Strict	Linear	LLaVA-OV, Qwen2.5-VL	$\leq 3.36\times$
Speech and Audio Models							
SSD	Independent	Small Audio LM	× (Distillation)	Relaxed	Linear	CosyVoice-2	$\approx 1.4\times$
SpecASR	Independent	Adaptive Len. + Tree	× (Tuning)	Strict	Tree	Llama, Vicuna	$3.04\times \sim 3.79\times$
SMUD	Independent	CTC Pseudo-Draft	✓	Relaxed	Linear	E-Branchformer ASR	$\approx 1.4\times$
Codec-MTP	Shared Backbone	Block Prediction	✓	Strict	Path	VALL-E, USLM	$4.0\times \sim 5.0\times$
UGSD	Independent	Edge-Cloud Gating	✓	Relaxed	Linear	Qwen2.5-Omni, Qwen3-Omni	$\approx 1.40\times$
CTC-SSD	Shared Backbone	CTC Encoder Draft	✓	Relaxed	Linear	Granite-Speech	$\leq 4.4\times$
Text-to-Image (Vision AR) Models							
LANTERN	Independent	Small AR Model	× (Tuning)	Relaxed	Linear	LlamaGen	$1.75\times \sim 1.82\times$
LANTERN++	Independent	Static Tree	× (Tuning)	Relaxed	Tree	LlamaGen	$\approx 2.56\times$
GSD	Independent	Dynamic Clustering	✓	Relaxed	Linear	AR Image Models	$\approx 3.8\times$
SJD	Shared Backbone	Jacobi Iteration	✓	Convergence	Iterative	LlamaGen, Emu3	$1.5\times \sim 2.0\times$
SJD^2	Shared Backbone	Denoise Trajectory	× (Tuning)	Convergence	Iterative	LlamaGen	$\approx 4.0\times$
MC-SJD	Shared Backbone	Maximal Coupling	✓	Convergence	Iterative	LlamaGen	$3.8\times \sim 4.2\times$
VVS	Independent	Confidence Skipping	✓	Strict	Linear	LlamaGen	$\approx 2.8\times$
SJD++	Shared Backbone	Token Reuse	× (Tuning)	Convergence	Iterative	LlamaGen	$\approx 2.4\times$
CSpD	Independent	Density-Ratio Sampling	✓	Relaxed	Path	MAR	$\leq 2.33\times$
SJD-PV	Shared Backbone	Phrase Library + Jacobi	✓	Relaxed	Path	Lumina-mGPT	$\leq 2.71\times$
MuLo-SD	Independent	Multi-Scale Drafting	× (Training)	Relaxed	Linear	Tar	$\leq 1.7\times$
Diffusion Models							
SpeCa	Drafter-Free Speculation	Feature Caching	✓	Relaxed	Path	DiT, FLUX, HunyuanVideo	$\leq 7.3\times$
Accel. Diff.	Drafter-Free Speculation	Coupling Jumps	✓	Relaxed	Path	DiT, EDM	$2.0\times \sim 3.0\times$
ASD	Drafter-Free Speculation	Stochastic Exchange	✓	Convergence	Path	DDPM	$1.8\times \sim 4.0\times$

Table 5: Systematic comparative summary of representative multimodal speculative decoding methods. Methods are grouped by modality and analyzed through the lens of our proposed two-stage taxonomy (Drafting and Verification). "✓" denotes tuning-free deployment, while "×" signifies the method requires auxiliary training, tuning, or distillation. Speedups are self-reported in the respective original papers under varying configurations (e.g., target models, hardware platforms, benchmarks, and batch sizes) and are therefore not directly comparable across methods. *Speculation-inspired reasoning framework, not canonical lossless speculative decoding acceleration. ‡Document-parsing VLM setting. †Block efficiency improvement, not direct walltime speedup.

mechanisms. Speech methods similarly span Independent (e.g., SSD (Lin et al., 2025b)) and Shared Backbone (e.g., Codec-MTP (Nguyen et al., 2025)) paradigms. Diffusion models uniquely introduce *Drafter-Free Speculation* approaches that generate trajectory segments rather than discrete tokens.

Second, this modality dependence is particularly visible in Text-to-Image (Vision AR) generation, where methods bifurcate along the drafting axis. Methods like LANTERN (Jang et al., 2025a) and GSD (So et al., 2025b) retain the dual-model framework but *relax acceptance criteria* in the visual latent space, which CSpD (Wang et al., 2024) adapts for continuous-valued representations. MuLo-SD (Peruzzo et al., 2026)

takes an orthogonal approach, combining a low-resolution independent drafter with multi-scale up-sampling to propose candidate patches beyond sequential token extension. In contrast, the SJD (Teng et al., 2025b) family eliminates separate drafters entirely through *Jacobi self-drafting*, and SJD-PV (Yu et al., 2026) further improves the acceptance rate by introducing phrase-level joint verification that exploits semantic continuity across consecutive visual tokens.

Third, *Tuning-free* adaptation remains a primary objective across all modalities (indicated by numerous ✓). Several tuning-free methods achieve substantial speedups through structural innovation, including MC-SJD (So et al., 2025a) (3.8–4.2×), EdgeSD (Huang et al., 2026) (3–5×), and GSD (So et al., 2025b) (≈ 3.8×). Nevertheless, closing the performance gap with trained drafters remains an open challenge: models like Spec-LLaVA (Huo et al., 2025) and DREAM (Hu et al., 2025b) still report the highest VLM speedups, both relying on dedicated distillation or tuning.

Finally, the choice of acceptance criterion tracks the output space. VLMs and Video–Language Models, which generate discrete tokens, overwhelmingly adopt exact-match verification. In contrast, modalities operating over continuous, latent, or perceptual outputs—VLA, Speech/Audio, Text-to-Image (Vision AR), and Diffusion Transformer (DiT)—more frequently rely on relaxed or convergence-based criteria to achieve practical acceptance rates.

6.2 Unified VLM Benchmarking

While Table 5 collects self-reported speedups under heterogeneous settings, a fair comparison requires controlled evaluation. MMSpec (Shen et al., 2026) provides the first standardized benchmark for speculative decoding in VLMs, evaluating vision-agnostic methods (EAGLE-1/2/3, Medusa) and vision-aware methods (MSD, ViSpec) across six subtasks on Qwen2.5-VL-7B and LLaVA-1.5-7B. The benchmark reveals that vision-agnostic methods can degrade below the autoregressive baseline on VLMs, while vision-aware methods consistently outperform them, confirming that modeling visual-conditioned token distributions is critical. Speedup also varies substantially across subtasks, highlighting the need for adaptive speculation strategies.

6.3 Other Modalities Benchmarking

The absence of similarly standardized benchmarks for Text-to-Image (Vision AR), VLA, Video, Speech, and Diffusion domains (§7) makes direct comparison across these modalities difficult, and the speedup figures in Table 5 remain incomparable because they reflect heterogeneous target models and evaluation protocols.

7 Open Challenges and Future Directions

While speculative decoding accelerates multimodal inference, applying the paradigm to high-dimensional multimodal spaces introduces several unsolved problems. Addressing these challenges requires advances in algorithm design, theoretical formulation, and hardware optimization.

The Multimodal Drafting Bottleneck. In standard LLM speculative decoding, drafting accounts for a negligible fraction of the total inference time. However, in multimodal models, even small drafter models must process high-resolution images, streaming audio, or long videos, resulting in a substantial "multimodal drafting bottleneck." As target models shift to complex "any-to-any" generation (e.g., interleaved text, image, and audio generation), building an ultra-lightweight drafter capable of generating reliable multi-format proposals becomes increasingly difficult. Future research should explore universally compressible representations or training-free self-speculation mechanisms (such as early-exiting or feature caching) to keep drafting overhead strictly bounded.

Extended Sequence Lengths and Memory Wall. The growth of sequence lengths in video understanding (VideoLMs) and high-fidelity audio generation places immense pressure on memory bandwidth. Although speculative decoding can alleviate parts of the memory-bandwidth bottleneck, longer KV caches and increasingly expensive attention memory accesses still cause memory transactions to overshadow arithmetic operations as sequence length grows. This highlights that VideoLMs fundamentally shift the bottleneck

toward memory-bound processing over long temporal sequences. While current compression strategies (§3.3.1) prune redundant visual tokens to mitigate this, excessive pruning risks degrading fine-grained semantic grounding. Future work should integrate KV cache quantization, offloading strategies, or linear-attention mechanisms directly into the speculative decoding verification phase.

Rigorous Verification Theory for Continuous Spaces. For many discrete-token generation settings, classical speculative decoding provides strong mathematical guarantees of lossless recovery (i.e., reproducing the exact output distribution of the target model). In contrast, models operating in continuous latent spaces, such as image and video Diffusion Transformers (DiTs) or visually quantized representations in visual autoregressive (VAR) models, lack equivalent theoretical bounds. Current verification strategies for these models (e.g., feature distance thresholds or semantic relaxations) rely heavily on empirical hyperparameter tuning. Developing a rigorous verification theory for continuous random variables that guarantees output distribution fidelity while allowing for extensive speculation is a critical open problem.

Strict Real-Time Constraints: Audio Streaming and VLA Control. Beyond latency reduction, Vision–Language–Action (VLA) models and speech systems are often deployed in environments where strict real-time constraints (e.g., high-frequency robotic control loops or time-to-first-audio) are non-negotiable. While speculative decoding increases overall throughput, naive block-level speculation can introduce unacceptable jitter or artificially delay the emission of the first acoustic or action frames. Designing drafters that generate structurally sound future actions or audio codecs far ahead of the target, without sacrificing streaming experience or suffering catastrophic rejection during physical execution, remains a key engineering challenge in real-time multimodal deployment.

Cross-Pollinating Representation-Level Verification to VLMs. Current VLM speculative decoding largely relies on strict token matching, forcing the drafter to predict the target's exact discrete token sequence. However, as demonstrated by several T2I (Vision AR) methods (§4.2.2), relaxing verification to the latent or representation space can substantially improve acceptance rates. A promising future direction is to adapt this *representation-level verification* to discrete VLM tokens. Because visual semantics are fundamentally continuous, verifying drafts based on embedding similarity or semantic equivalence rather than exact lexical matches may help overcome current tuning-free VLM speedup limitations without requiring costly self-correction.

Algorithm–Hardware Co-Design. Advanced speculative decoding algorithms frequently employ dynamic, tree-structured speculation to verify multiple candidate trajectories simultaneously (§4.1.2). This dynamic branching creates irregular compute geometries, sparse attention masks, and dynamic batch sizes, which underutilize modern AI accelerators (e.g., GPUs and TPUs) optimized for static, dense matrix multiplications. To unlock the theoretical speedups of width-parallel speculative decoding, the field needs hardware–algorithm co-design, including specialized CUDA kernels for sparse tree-attention and optimized memory access patterns tailored for parallel target verification.

Evaluation Standardization and Reproducibility. The speculative decoding literature currently suffers from fragmented evaluation. Reported speedup multipliers vary widely depending on the baseline implementation (e.g., vanilla PyTorch vs. vLLM/TensorRT), hardware platform, batch size, and prompt characteristics. Unlike text generation where speedup is easily quantified by tokens per second, multimodal generation speedup depends on the input length (e.g., the number of images/frames relative to the generated text). While recent efforts such as MMSpec (Shen et al., 2026) have introduced standardized benchmarks for vision–language models (§6.2), no equivalent evaluation platform exists for Vision–Language–Action control, Video–Language understanding, Speech/Audio models, Text-to-Image (Vision AR) generation, or Diffusion Transformers. This cross-modal benchmarking gap hinders fair comparison of algorithmic contributions and obscures which innovations genuinely transfer across modalities. Establishing standardized cross-modal benchmarking frameworks that isolate algorithmic efficiency from system-level engineering and define modality-appropriate quality metrics is essential for transparent and reproducible progress.

8 Conclusion

This survey presented a comprehensive, unified taxonomy of speculative decoding for multimodal models. Moving beyond the text-centric roots of the paradigm, we systematically analyzed draft architecture, execution strategies, optimization patterns, verification criteria, and inference framework support across six distinct domains: Vision–Language Models (VLMs), Vision–Language–Action (VLA) agents, Video–Language models, Speech systems, Text-to-Image (Vision Auto-Regressive, T2I) generators, and Diffusion-based generators. By formalizing the problem space, we identified key recurring design patterns, including visual token compression, KV cache optimization, target-informed transfer, drafter-target alignment, verification cost reduction, relaxed acceptance criteria, and verify-to-draft feedback loops.

Each modality introduces unique computational bottlenecks that demand tailored solutions, from extensive spatial pruning and multi-scale drafting in text-to-image synthesis to continuous-space and phrase-level verification criteria, strict real-time latency constraints in robotic control, and tolerance for stochastic variance in diffusion processes. As foundation models increasingly adopt native multimodal generation, sequential and iterative inference latency is becoming a critical deployment constraint. Recent standardized VLM benchmarks suggest that text-only speculation can degrade substantially on vision–language tasks, underscoring the importance of the domain-aware techniques reviewed herein. Speculative decoding provides a promising pathway toward interactive-latency AI systems without sacrificing generative quality. This survey provides a technical foundation and roadmap for researchers and practitioners working to accelerate multimodal inference.

References

Divya Jyoti Bajpai and Manjesh Kumar Hanawal. FastVLM: Self-speculative decoding for fast vision-language model inference. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 1166–1183, 2025.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. *arXiv preprint arXiv:2501.05370*, 2025.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Haotian Dong, Ye Li, Rongwei Lu, Chen Tang, Shu-Tao Xia, and Zhi Wang. Vvs: Accelerating speculative decoding for visual autoregressive generation via partial verification skipping. *arXiv preprint arXiv:2511.13587*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Mukul Gagrani, Raghavv Goel, Wonseok Jeon, Junyoung Park, Mingu Lee, and Christopher Lott. On speculative decoding for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8285–8289, 2024.

Mugilan Ganesan, Shane Segal, Ankur Aggarwal, Nish Sinnadurai, Sean Lie, and Vithursan Thangarasa. MASSV: Multimodal adaptation and self-data distillation for speculative decoding of vision-language models. *arXiv preprint arXiv:2505.10526*, 2025.

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

Hengyuan Hu, Aniket Das, Dorsa Sadigh, and Nima Anari. Diffusion models are secretly exchangeable: Parallelizing DDPMs via autospeculation. *arXiv preprint arXiv:2505.03983*, 2025a.

Yunhai Hu, Tianhua Xia, Zining Liu, Rahul Raman, Xingyu Liu, Bo Bao, Eric Sather, Vithursan Thangarasa, and Sai Qian Zhang. Dream: Drafting with refined target features and entropy-adaptive cross-attention fusion for multimodal speculative decoding. *arXiv preprint arXiv:2505.19201*, 2025b.

Haiduo Huang, Fuwei Yang, Zhenhua Liu, Xuanwu Yin, Dong Li, Pengju Ren, and Emad Barsoum. Specvlm: Fast speculative decoding in Vision-Language models. *arXiv preprint arXiv:2509.11815*, 2025.

Hualong Huang, Wenhan Zhan, Hancong Duan, Kai Peng, Geyong Min, Zijia Zhao, Zitian Zhao, and Yalan Ye. Edgesd: Efficient speculative decoding with vision-decoding disaggregation for MLLM inference in edge-cloud networks. *IEEE Transactions on Mobile Computing*, 2026.

Hugging Face. Text generation inference: A rust, python and gRPC server for text generation inference. `https://github.com/huggingface/text-generation-inference`, 2023. GitHub repository, accessed April 17, 2026.

Mingxiao Huo, Jiayi Zhang, Hewei Wang, Jinfeng Xu, Zheyu Chen, Huilin Tai, and Yijun Chen. Spec-LLaVA: Accelerating vision-language models with dynamic tree-based speculative decoding. *arXiv preprint arXiv:2509.11961*, 2025.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. LANTERN: Accelerating visual autoregressive models with relaxed speculative decoding. *arXiv preprint arXiv:2410.03355*, 2025a. ICLR 2025.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. LANTERN++: Enhanced relaxed speculative decoding with static tree drafting for visual auto-regressive models. *arXiv preprint arXiv:2502.06352*, 2025b. ICLR 2025 SCOPE Workshop.

Yicheng Ji, Jun Zhang, Heming Xia, Jinpeng Chen, Lidan Shou, Gang Chen, and Huan Li. Specvlm: Enhancing speculative decoding of video llms via verifier-guided token pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7216–7230, 2025.

Jialiang Kang, Han Shu, Wenshuo Li, Yingjie Zhai, and Xinghao Chen. Vispec: Accelerating vision-language models with vision-aware speculative decoding. *arXiv preprint arXiv:2509.15235*, 2025.

Quan Kong, Yuhao Shen, Yicheng Ji, Huan Li, and Cong Wang. Parallelvlm: Lossless video-llm acceleration with visual alignment aware parallel speculative decoding. *arXiv preprint arXiv:2603.19610*, 2026.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Minjae Lee, Wonjun Kang, Byeongkeun Ahn, Christian Classen, Minghao Yan, Hyung Il Koo, and Kangwook Lee. In-batch ensemble drafting: Robust speculative decoding for lvlms. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.

Minjae Lee, Wonjun Kang, Byeongkeun Ahn, Christian Classen, Kevin Galim, Seunghyuk Oh, Minghao Yan, Hyung Il Koo, and Kangwook Lee. Tabed: Test-time adaptive ensemble drafting for robust speculative decoding in lvlms. *arXiv preprint arXiv:2601.20357*, 2026.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Gen Li and Peiyu Liu. Fastv-rag: Towards fast and fine-grained video qa with retrieval-augmented generation. *arXiv preprint arXiv:2601.01513*, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Shenggui Li, Yikai Zhu, Chao Wang, Fan Yin, Shuai Shi, Yubo Wang, Yi Zhang, Yingyi Huang, Haoshuai Zheng, and Yineng Zhang. SpecForge: Train speculative decoding models effortlessly. `https://github.com/sgl-project/specforge`, 2025a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 7421–7432, 2024b.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025b.

Wenhui Liao, Hongliang Li, Pengyu Xie, Xinyu Cai, Yufan Shen, Yi Xin, Qi Qin, Shenglong Ye, Tianbin Li, Ming Hu, et al. Training-free acceleration for document parsing vision-language model with hierarchical speculative decoding. *arXiv preprint arXiv:2602.12957*, 2026.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

Luxi Lin, Zhihang Lin, Zhanpeng Zeng, and Rongrong Ji. Speculative decoding reimagined for multimodal large language models. *arXiv preprint arXiv:2505.14260*, 2025a.

Zijian Lin, Yang Zhang, Yougen Yuan, Yuming Yan, Jinjiang Liu, Zhiyong Wu, Pengfei Hu, and Qun Yu. Accelerating autoregressive speech synthesis inference with speech speculative decoding. *arXiv preprint arXiv:2505.15380*, 2025b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Fei Ren, Shaobo Wang, Kaixin Li, and Linfeng Zhang. Speca: Accelerating diffusion transformers with speculative feature caching. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10024–10033, 2025a.

Yuhan Liu, Lianhui Qin, and Shengjie Wang. Small drafts, big verdict: Information-intensive visual reasoning via speculation. *arXiv preprint arXiv:2510.20812*, 2025b.

Qitan Lv, Tianyu Liu, Wen Wu, Xuenan Xu, Bowen Zhou, Feng Wu, and Chao Zhang. Hippo: Accelerating video large language models inference via holistic-aware parallel speculative decoding. *arXiv preprint arXiv:2601.08273*, 2026.

MMRazor and MMDeploy Teams. Lmdeploy: A toolkit for compressing, deploying, and serving LLMs. `https://github.com/InternLM/lmdeploy`, 2023. GitHub repository, accessed March 11, 2026.

Tan Dat Nguyen, Ji-Hoon Kim, Jeongsoo Choi, Shukjae Choi, Jinseok Park, Younglo Lee, and Joon Son Chung. Accelerating codec-based speech synthesis with multi-token prediction and speculative decoding. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

NVIDIA. Tensorrt-LLM. https://github.com/NVIDIA/TensorRT-LLM, 2024. Accessed: 2026-03-11.

NVIDIA. NVIDIA Dynamo: A datacenter scale distributed inference serving framework. https://github.com/ai-dynamo/dynamo, 2025. GitHub repository, accessed April 17, 2026.

NVIDIA. TensorRT-Edge-LLM: High-performance inference for LLMs and VLMs on embedded platforms. https://github.com/NVIDIA/TensorRT-Edge-LLM, 2026. GitHub repository, accessed March 11, 2026.

Koji Okabe and Hitoshi Yamamoto. Simultaneous masked and unmasked decoding with speculative decoding masking for fast ASR without accuracy loss. In *Proc. Interspeech 2025*, pp. 634–638, 2025.

Elia Peruzzo, Guillaume Sautière, and Amirhossein Habibian. Multi-scale local speculative decoding for image generation. *arXiv preprint arXiv:2601.05149*, 2026.

Red Hat. Speculators: A unified library for speculative decoding algorithms in LLM serving. https://github.com/vllm-project/speculators, 2025.

George Saon, Samuel Thomas, Takashi Fukuda, Tohru Nagano, Avihu Dekel, and Luis Lastras. Self-speculative decoding for llm-based asr with ctc encoder drafts. *arXiv preprint arXiv:2603.11243*, 2026.

Hui Shen, Xin Wang, Ping Zhang, Yunta Hsieh, Qi Han, Zhongwei Wan, Ziheng Zhang, Jingxuan Zhang, Jing Xiong, Ziyuan Liu, et al. MMSpec: Benchmarking speculative decoding for vision-language models. *arXiv preprint arXiv:2603.14989*, 2026.

Junhyuk So, Hyunho Kook, Chaeyeon Jang, and Eunhyeok Park. Mc-sjd: Maximal coupling speculative jacobi decoding for autoregressive visual generation acceleration. *arXiv preprint arXiv:2510.24211*, 2025a.

Junhyuk So, Juncheol Shin, Hyunho Kook, and Eunhyeok Park. Grouped speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2508.07747*, 2025b. ICCV 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Qwen Team. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Yao Teng, Zhihuan Jiang, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. SJD++: Improved speculative jacobi decoding for training-free acceleration of discrete auto-regressive text-to-image generation. *arXiv preprint arXiv:2512.07503*, 2025a.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating auto-regressive Text-to-Image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*, 2025b. ICLR 2025.

Yao Teng, Fuyun Wang, Xian Liu, Zhekai Chen, Han Shi, Yu Wang, Zhenguo Li, Weiyang Liu, Difan Zou, and Xihui Liu. Speculative jacobi-denoising decoding for accelerating autoregressive text-to-image generation. *arXiv preprint arXiv:2510.08994*, 2025c. NeurIPS 2025.

Yujia Tong, Tian Zhang, Yunyang Wan, Kaiwei Lin, Jingling Yuan, and Chuang Hu. Sage: Accelerating Vision-Language models via entropy-guided adaptive speculative decoding. *arXiv preprint arXiv:2602.00523*, 2026.

Hanzhen Wang, Jiaming Xu, Jiayi Pan, Yongkang Zhou, and Guohao Dai. Specprune-VLA: Accelerating vision-language-action models via action-aware self-speculative pruning. *arXiv preprint arXiv:2509.04043*, 2025a.

Songsheng Wang, Rucheng Yu, Zhihang Yuan, Chao Yu, Feng Gao, Yu Wang, and Derek F. Wong. Spec-VLA: Speculative decoding for vision-language-action models with relaxed acceptance. *arXiv preprint arXiv:2507.22424*, 2025b.

Zihua Wang, Ruibo Li, Haozhe Du, Joey Tianyi Zhou, Yu Zhang, and Xu Yang. Flash: Latent-Aware Semi-Autoregressive speculative decoding for multimodal tasks. *arXiv preprint arXiv:2505.12728*, 2025c.

Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. Continuous speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2411.11925*, 2024.

Linye Wei, Shuzhang Zhong, Songqiang Xu, Runsheng Wang, Ru Huang, and Meng Li. Specasr: Accelerating LLM-based automatic speech recognition via speculative decoding. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7. IEEE, 2025.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7655–7671, 2024.

Zhinan Xie, Peisong Wang, Shuang Qiu, and Jian Cheng. Hivis: Hiding visual tokens from the drafter for speculative decoding in Vision-Language models. *arXiv preprint arXiv:2509.23928*, 2025.

Xiangyuan Xue, Jiajun Lu, Yan Gao, Gongping Huang, Ting Dang, and Hong Jia. Edge–cloud collaborative speech emotion captioning via token-level speculative decoding in audio-language models. *arXiv preprint arXiv:2603.11397*, 2026.

Chaoqun Yang, Ran Chen, Muyang Zhang, Weiguang Pang, Yuzhi Chen, Rongtao Xu, Kexue Fu, Changwei Wang, and Longxiang Gao. Aasd: Accelerate inference by aligning speculative decoding in multimodal large language models. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, pp. 1–7. IEEE, 2025.

Zhehao Yu, Baoquan Zhang, Bingqi Shan, Xinhao Liu, Dongliang Zhou, Guotao Liang, Guangming Ye, and Yunming Ye. Sjd-pv: Speculative jacobi decoding with phrase verification for autoregressive image generation. *arXiv preprint arXiv:2603.06666*, 2026.

Libo Zhang, Zhaoning Zhang, Wangyang Hong, Peng Qiao, and Dongsheng Li. Sparrow: Text-anchored window attention with visual-semantic glimpsing for speculative decoding in video LLMs. *arXiv preprint arXiv:2602.15318*, 2026.

Xuan Zhang, Cunxiao Du, Sicheng Yu, Jiawei Wu, Fengzhuo Zhang, Wei Gao, and Qian Liu. Sparse-to-dense: A free lunch for lossless acceleration of video understanding in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 734–742, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody H Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583, 2024.

Zihao Zheng, Zhihao Mao, Maoliang Li, Jiayu Chen, Xinhao Sun, Zhaobo Zhang, Donggang Cao, Hong Mei, and Xiang Chen. KERV: Kinematic-rectified speculative decoding for embodied VLA models. In *Proceedings of the 63rd ACM/IEEE Design Automation Conference (DAC)*, 2026a.

Zihao Zheng, Zhihao Mao, Sicheng Tian, Maoliang Li, Jiayu Chen, Xinhao Sun, Zhaobo Zhang, Xuanzhe Liu, Donggang Cao, Hong Mei, et al. Heisd: Hybrid speculative decoding for embodied vision-language-action models with kinematic awareness. *arXiv preprint arXiv:2603.17573*, 2026b.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.