Preprints.org

Article

Not peer-reviewed version

Speculative Decoding for Multimodal Models: A Survey

Yifan Zhang [*,†] , Yuren Wang [†] , Yunta Hsieh , Xin Wang , Ping Zhang , Ziyi Yang , Jianing Ma , Zesen Zhao , Boyuan Zheng , Hei Ting (Una) Chan , Jiarui Li , Xueshen Liu , Kunxiao Gao , Ruiyao Liu , Jingxuan Zhang , Junchen Li , Zhongwei Wan , Ziheng Zhang , Jing Xiong , Shatong Zhu , Hangrui Cao , Hui Shen [*]

Posted Date: 30 March 2026

doi: 10.20944/preprints202603.2344.v1

Keywords: speculative decoding; multimodal models; vision-language models; text-to-image models; vision-language-action models; video-language models; speech and audio models; diffusion models

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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

Article

Speculative Decoding for Multimodal Models: A Survey

Yifan Zhang [1,*,†] Yuren Wang [2,†] Yunta Heish [1], Xin Wang [3], Ping Zhang [3], Ziyi Yang [4], Jianing Ma [5], Zesen Zhao [1], Boyuan Zheng [1], Hei Ting (Una) Chan [1], Jiarui L [1], Xueshen Liu [1], Kunxiao Gao [1], Ruiyao Liu [6], Jingxuan Zhang [3], Junchen Li [7], Zhongwei Wan [3], Ziheng Zhang [8], Jing Xiong [9], Shatong Zhu [10], Hangrui Cao [11] and Hui Shen [1,*]

[1] University of Michigan, USA
[2] University of Sussex, UK
[3] The Ohio State University, USA
[4] The University of British Columbia, Canada
[5] Northern Arizona University, USA
[6] University of Pennsylvania, USA
[7] City University of Hong Kong, Hong Kong
[8] The University of Texas at Austin, USA
[9] The University of Hong Kong, Hong Kong
[10] Stanford University, USA
[11] Carnegie Mellon University, USA
[†] These authors contributed equally to this work.
[*] Correspondence: yifanzhg@umich.edu (Y.Z.); huishen@umich.edu (H.S.)

Abstract

Multimodal generative models have demonstrated remarkable capabilities in visual understanding, audio synthesis, and embodied control. Such capabilities, however, come with substantial inference overhead due to autoregressive decoding or iterative generation processes, compounded by modality-specific challenges including extensive visual token redundancy, strict real-time latency 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, Text-to-Image, Vision–Language–Action, Video–Language, Speech, and Diffusion models. We organize the literature in a unified taxonomy consisting of 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 design patterns, including token compression, target-informed transfer, and relaxed acceptance, 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 have also created a GitHub repository where we organize the papers featured in this survey at https://github.com/zyfzs0/Multimodal-Models-Speculative-Decoding-Survey, and will actively maintain it to incorporate new research as it emerges. We hope this survey can serve as a valuable resource for researchers and practitioners working on accelerating multimodal inference.

Keywords: speculative decoding; multimodal models; vision-language models; text-to-image models; vision-language-action models; video-language models; speech and audio models; diffusion models

1. Introduction

While multimodal generative models have achieved high fidelity and accuracy in complex reasoning and synthesis tasks, their deployment is severely limited by the sequential nature of autoregressive

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

generation. High-resolution images, long video streams, and high-frequency robotic control loops 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; Text-to-Image (T2I) models suffer from prolonged layer-by-layer or token-by-token generation times; and Vision–Language–Action (VLA) models face strict real-time control latency limits. As foundation models scale, this autoregressive latency remains a primary obstacle to practical deployment.

Speculative decoding accelerates autoregressive generation without degrading output quality [1–3]. By decomposing generation into two stages, a lightweight *draft* model efficiently proposing K candidate tokens and a full-capacity *target* model verifying these tokens in parallel, the 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 [3]. 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), Text-to-Image (T2I) systems, Vision–Language–Action (VLA) agents, Video–Language models, Speech systems, and Diffusion-based generators are siloed within their respective sub-communities, lacking a unified framework.

The goal of this survey is to provide a unified overview of speculative decoding for multimodal models. As illustrated in Figure 2, we organize the literature in a unified taxonomy consisting of two primary axes, covering the *draft generation stage* and the *verification and acceptance stage*, complemented by an analysis of inference framework support. These two axes cover distinct yet interconnected research topics, collectively providing a systematic and comprehensive review 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 verification 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). We hope this survey can serve as a valuable resource for researchers and practitioners working on accelerating multimodal inference.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

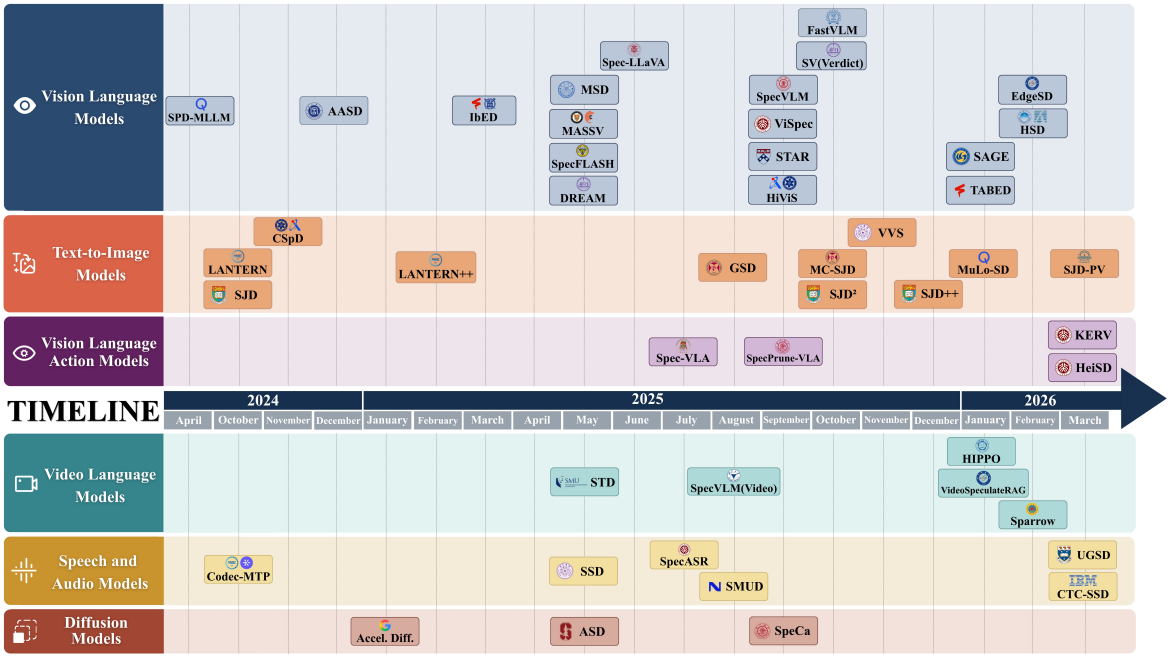

Figure 1. Timeline of multimodal speculative decoding methods surveyed in this work. Colors indicate modality: methods span Vision–Language, Text-to-Image, Vision–Language–Action, Video–Language, Speech, and Diffusion models.

2. Background

2.1. Standard Speculative Decoding

Standard speculative decoding accelerates autoregressive inference by decomposing generation into a *draft-then-verify* paradigm [1–3]. 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 the final output distribution mathematically matches $p_{\text{target}}(x \mid x_{<t})$ losslessly.

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 sub-optimal $\mathbb{E}[\alpha]$ 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) with large language models [4,5]. 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

Text-to-Image Models (T2I)

synthesize images either by predicting discrete codebook indices within an autoregressive transformer [6] or through latent diffusion. Both paradigms suffer from prolonged generation times due to either long flattened token sequences or sequential iterative denoising steps.

Vision–Language–Action Models (VLAs)

map visual input and textual instructions directly to low-level robotic control tokens [7]. 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 [8]. 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 [9]. Because human hearing processes audio semantically, these models exhibit *perceptual tolerance*: acoustically equivalent sounds can have differing discrete representations, violating the premise of Equation (1) and requiring relaxed verification criteria.

Diffusion Models

generate structured outputs by iteratively removing noise in continuous latent spaces [10,11]. 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.

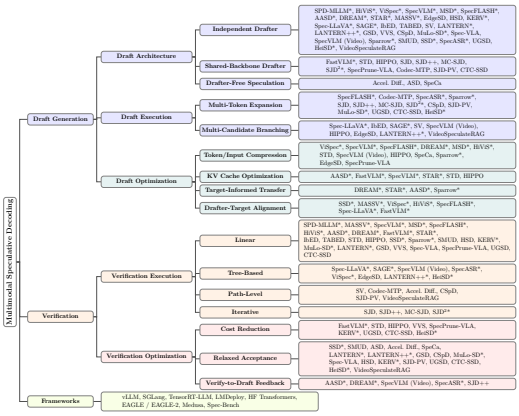

Figure 2. Taxonomy of speculative decoding for multimodal models. * denotes methods requiring training/fine-tuning; unmarked methods are training-free.
Blue : draft architecture & execution. Teal : draft optimization. Orange : verification execution. Red : verification optimization. Green : frameworks.

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 the *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 2 into three dimensions: draft architecture (§3.1), draft execution (§3.2), and draft optimization (§3.3).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

3.1. Draft Architecture

Figure 3 presents the sub-taxonomy of draft architecture, and Table 1 summarizes their formulations.

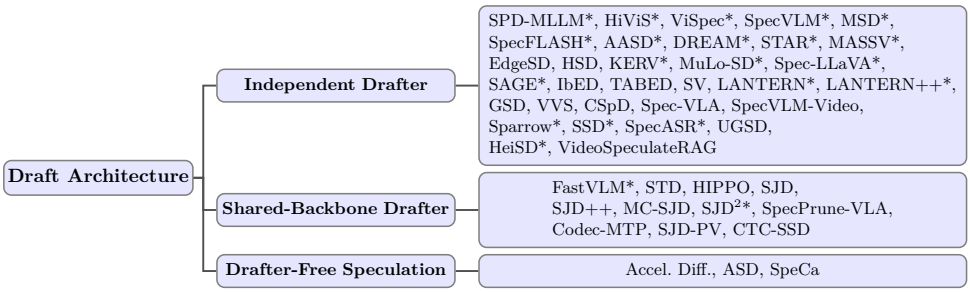

Figure 3. Sub-taxonomy of draft architecture (§3.1). Methods are categorized by their structural relationship to the target model. * denotes methods requiring training/fine-tuning.

Table 1. Summary of formulations for draft architecture in multimodal speculative decoding. $p_{1...K}$ 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), $\tilde{z}_{1...K}$ 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.

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

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 [12] 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. Note that speculative decoding does not require the drafter to understand images; only the target must verify correctness.

Multimodal Independent Drafters.

To recover the performance drop 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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

ViSpec [13] observes that shallow drafters cannot replicate the multi-layer visual filtering that large VLMs perform, so the drafter needs a more efficient visual entry point. The method introduces a Q-Former-style [14] vision adaptor: learnable query tokens attend to the original visual features via cross-attention, producing a small set of compressed visual tokens and a global visual vector injected at every text position.

SpecFLASH [15] combines latent-aware visual token compression and semi-autoregressive drafting (detailed in §3.2.1), allowing it to propose blocks of candidate tokens at once.

HiViS [16] removes explicit visual token processing from the drafter entirely. Instead, it conditions on a precomputed semantic embedding exported by the target model, augmented with step-aware residual vectors. This design eliminates drafter-side visual encoding while preserving multimodal grounding.

SpecVLM [17] addresses the VLM prefill bottleneck caused by visual token volume through an elastic visual compressor supporting multiple adaptive compression modes. Training uses online logit distillation to ensure distribution-level compatibility between drafter and target.

AASD [18] employs a dedicated speculative module that attends to the target's KV cache through learned projections, compressing the multimodal KV before cross-attention.

DREAM [19] conditions the drafter on selected target intermediate features via entropy-adaptive cross-attention, selectively injecting features from the most informative target layers.

STAR [20] reframes drafter design as a neural architecture search (NAS) problem [21], discovering the optimal drafter structure including visual token count, attention heads, and target feature interaction layers.

MASSV [22] trains a compact independent LM as the drafter and equips it with a lightweight multimodal projector. Training follows a two-stage protocol: projector pretraining on paired image-text data followed by self-data distillation from the target VLM.

IbED [23] (In-batch Ensemble Drafting) discovers that no single drafting strategy dominates across all inputs. Rather than choosing one, it runs multiple independent strategies as a batch, combining their probability distributions and exploiting the near-zero marginal latency of batch inference on small drafters. TABED [24] extends this ensemble paradigm with test-time adaptation: at each drafting block it samples a set of candidate ensemble weights over multimodal and text-only drafts, then selects the weights minimizing the KL divergence (or maximizing token matches) against the target distribution over a past time-step window, effectively performing test-time adaptation using the ground-truth labels available through verification. The method is fully training-free and applicable across diverse VLM input scenarios.

MSD [25] decouples text and vision in the drafting pipeline, processing each modality according to its characteristics.

EdgeSD [26] introduces a *vision-decoding disaggregation* (VED) architecture for edge-cloud VLM speculative decoding. VED decouples the visual encoder and LLM backbone of the draft VLM across separate edge servers, connected via an asynchronous migration mechanism with a Joint-Shortest-Queue scheduler, overcoming single-node memory constraints. EdgeSD further contributes a bandwidth-aware dynamic image token merging (ITM) method and an adaptive token tree via delta-stepping (§3.2.2, §3.3.1).

HSD [27] (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 (layout analysis + OCR) serves as the draft model, and the page is partitioned into semantically independent regions whose draft–verify cycles execute in parallel. A second page-level verification stage reassembles all accepted region outputs against the full VLM. HSD introduces a tolerance-based acceptance criterion (τ-matching, §4.2.2) that permits minor formatting variations.

Visual Domain Independent Drafters.

LANTERN [28] and LANTERN++ [29] train compact visual AR models as drafters for image generation, retaining the standard dual-model framework while introducing relaxed latent-space

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

acceptance to overcome the low token-match rates associated with visual codebook prediction (§4.2.2). GSD [30] pairs an independent drafter with dynamic token clustering, grouping visually equivalent tokens to boost acceptance without training. VVS [31] and CSpD [32] similarly employ independent drafters but shift their innovations to verification skipping and continuous-density acceptance, respectively. MuLo-SD [33] (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. Its local spatial verification strategy is discussed in §4.2.2.

Vision–Language–Action Independent Drafters.

Spec-VLA [34] 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 (§4.2.2).

KERV [35] (Kinematic-Rectified Speculative Decoding) further advances VLA speculation by combining token-domain drafting with kinematic-domain compensation. KERV employs a lightweight autoregressive draft model. When verification rejects a draft token, instead of the costly re-inference step, KERV activates a Kalman Filter (KF) that predicts the remaining actions in the current action slice from the robot's kinematic history, avoiding GPU-side recomputation entirely. A kinematic-based threshold adjustment strategy dynamically tunes the acceptance threshold based on real-time kinematic variability, replacing the static threshold used by Spec-VLA (§4.2.2).

HeiSD [36] introduces *hybrid* speculative decoding for VLA models, dynamically switching between retrieval-based and drafter-based SD depending on 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 two key optimizations. First, an adaptive verify-skip mechanism selectively bypasses verification when feature similarity to historical trajectories is high (§4.2.1). Second, a sequence-wise relaxed acceptance strategy groups kinematically correlated tokens (e.g., position XYZ, rotation angles) and accepts the entire sequence when its aggregate bias remains small (§4.2.2).

Video–Language Independent Drafters.

Sparrow [37] targets long-video scenarios where long visual token sequences cause attention dilution. It offloads visual computation to the target model and eliminates the drafter's visual KV cache, relying on target-informed knowledge transfer and attention constraints (detailed in §3.3).

VideoSpeculateRAG [38] introduces speculative decoding into the video retrieval-augmented generation (RAG) pipeline. Rather than concatenating all retrieved documents into a single long-context input, each retrieved document is independently paired with the video and question and processed in parallel by a lightweight draft VLM, producing multiple candidate answers simultaneously. A larger verifier then scores each candidate through a two-stage process: δ-tolerant reliability filtering followed by entity-alignment reranking (§4.1.3).

Speech Independent Drafters.

Speech speculative decoding exploits the low-entropy, strongly conditioned nature of acoustic generation. SSD [39] (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 [40] applies a related paradigm to automatic speech recognition, pairing a lightweight draft ASR model with the full target recognizer. SMUD [41] 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

then refined via a single non-autoregressive decoder forward pass, acting as an efficient one-shot pseudo-draft step. UGSD [42] (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 (§4.2.2), and the block length adapts dynamically based on recent acceptance history.

The architectural mechanisms through which these drafters receive target-derived signals, such as feature injection, KV cache sharing, and hidden state reuse, are systematically analyzed as draft optimization patterns in §3.3.3.

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.

FastVLM [43] 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. An imitation network bridges the two stages: it takes the n-th layer output as input and learns to mimic the behavior of the remaining $L - n$ layers, with the backbone frozen so only the imitation network is trainable. Because draft and verification share the same weights and layer ordering, computation from the first n layers transfers directly to the verification pass. Rejected tokens are corrected by the full model and fed back to improve the drafter through iterative imitation learning.

In the video domain, STD [44] 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.

The SJD family [45–47] applies this self-drafting philosophy to visual autoregressive generation through Jacobi-style parallel prediction. SJD [45] 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++ [46] reuses high-confidence tokens across iterations to accelerate convergence, while MC-SJD [47] stabilizes convergence via maximal coupling. SJD-PV [48] 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 (§4.1.3). These methods share the backbone philosophy of FastVLM [43] and STD [44] but differ in mechanism: rather than layer-splitting or KV-splitting, they exploit iterative fixed-point convergence of the full model.

In the VLA domain, SpecPrune-VLA [49] 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 video domain, HIPPO [50] 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 discussed under token compression (§3.3.1). In the speech domain, Codec-MTP [51] 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 [52] (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, determined by frame-level CTC entropy, are accepted directly without invoking the LLM at all. When the entropy exceeds a threshold, the CTC hypothesis is verified in a single LLM forward pass under a relaxed token-likelihood acceptance criterion (§4.2.2); if verification fails, AR decoding

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

resumes from the accepted prefix. This three-stage pipeline simultaneously improves WER (through complementary CTC-LLM error patterns) and accelerates inference.

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.

Coupling-Based Proposals.

ACCELERATED DIFFUSION SAMPLING [53] 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.

Exchangeability-Based Parallel Proposals.

ASD [54] 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.

Feature-Level Forecasting.

SPECA [55] 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

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 before verification. The two sub-categories 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. Figure 4 illustrates the draft execution sub-taxonomy, and Table 2 contrasts the two strategies.

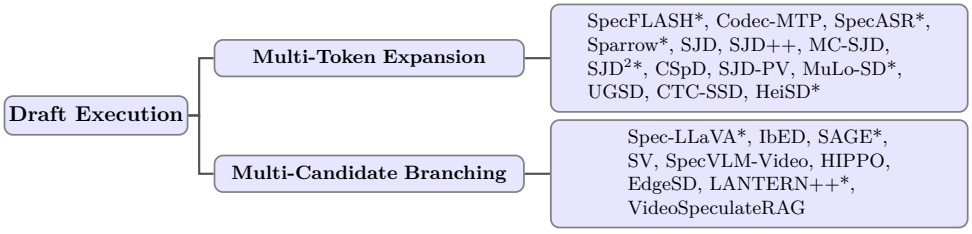

Figure 4. Sub-taxonomy of draft execution strategies (§3.2). Multi-token expansion explores depth; multi-candidate branching explores width. * denotes methods requiring training/fine-tuning.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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.

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

3.2.1. Multi-Token Expansion

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

SpecFLASH [15] 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 [18] and FastVLM [43] 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. The SJD family [45–47,56] achieves depth parallelism natively through its Jacobi iteration framework (§3.1.2): 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 [56] further structures each Jacobi round as a denoising trajectory from Gaussian noise. CSpD [32] 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 [33] 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.

In the video domain, Sparrow [37] 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 [51] predicts n future codec tokens in a single forward pass, reducing decoding steps proportionally. SpecASR [40] adaptively extends draft length, dynamically adjusting based on prediction confidence to minimize verification rounds.

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.

Spec-LLaVA [57] constructs a dynamic speculation tree with online structural and budget pruning. Structural pruning removes low-probability or redundant branches; budget pruning limits the tree to the top-n candidate tokens, preventing tree explosion. The target model then verifies this tree structure (detailed in §4.1.2).

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

SV (Speculative Verdict) [58] operates at the reasoning level rather than the token level. Multiple lightweight VLMs independently generate diverse reasoning paths in the draft stage. A consensus filter based on negative log-likelihood scores, measuring cross-model agreement ("how likely would other models find this answer?"), selects high-agreement paths. The verdict model then processes these paths not as candidates to vote on but as *evidence* to synthesize: in a single inference call, it reasons over all paths and generates a new final answer, functioning as an evidence synthesizer rather than a voter.

SAGE [59] dynamically adjusts the speculation tree shape using the drafter's prediction entropy at each step. High entropy indicates uncertain predictions, prompting wider branching; low entropy allows deeper speculation. This adaptive width/depth allocation optimizes the tree structure per step rather than using a fixed topology.

In the video domain, SpecVLM (Video) [60] adopts an EAGLE-style [61] static tree structure with tree attention masks, supporting direct integration with tree-based speculative decoding. The draft model generates multi-branch candidate trees over pruned video tokens, which the target verifies via structured tree attention.

VideoSpeculateRAG [38] takes a document-parallel branching approach: rather than constructing a token-level tree, it generates one complete candidate answer per retrieved document, treating each document–video–question triple as an independent branch. The verifier selects the best branch through path-level two-stage scoring (§4.1.3).

LANTERN++ [29] applies static tree drafting to visual autoregressive generation, constructing a fixed tree topology that enables multi-branch speculation over visual codebook tokens with relaxed latent-space acceptance across all branches.

HIPPO [50] does not expand a conventional multi-branch candidate tree; instead, it realizes parallelism through pipelined overlap of draft and verification across time steps. Its "parallel speculative execution" overlaps draft generation of batch $t+1$ with target verification of batch t, transforming the serial draft-then-verify cycle into a production-line pipeline.

EdgeSD [26] formulates adaptive token 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. Unlike fixed tree structures, EdgeSD [26]'s tree depth and branching configuration adapt per input sequence.

3.3. Draft Optimization

As shown in Figure 5, 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, T2I, VLA, video, and speech systems, where high-dimensional inputs introduce substantial redundancy and require careful draft–target coordination.

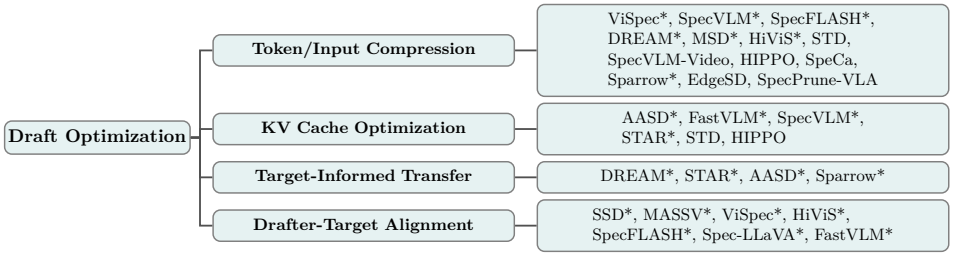

Figure 5. Sub-taxonomy of draft optimization strategies (§3.3). These strategies recur across modalities. Target-Informed Transfer covers inference-time signal injection; Drafter-Target Alignment covers training-time compatibility. * denotes methods requiring training/fine-tuning.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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.

Semantic Visual Compression.

ViSpec [13] uses a Q-Former-style [14] adaptor to compress visual tokens into a small set of queries plus a global vector (see §3.1.1 for architectural detail). HiViS [16] eliminates visual-prefill cost entirely, replacing visual tokens with fused semantic embeddings from the target model. SpecFLASH [15] applies latent-aware compression using target sub-top-layer features (architectural detail in §3.2.1).

Adaptive and Dynamic Compression.

SpecVLM [17] adaptively applies pooling, convolution, or pruning based on token redundancy, supporting multiple compression modes within a single framework. DREAM [19] uses target intermediate features to guide visual token selection, retaining only informative tokens. MSD [25] decouples text and vision processing, effectively bypassing visual tokens during drafting. EdgeSD [26] 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.

A recurring pattern emerges across modalities where attention scores guide token selection. STD [44] selects the top-K visual KV cache entries per layer and per head based on prefill-stage attention scores for VideoLMs. SpecVLM (Video) [60] 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 [50] fuses three complementary scoring signals (global semantic relevance, temporal redundancy, and spatial redundancy) into a holistic score for video token selection. SpecPrune-VLA [49] 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.

Sparrow [37] takes the most radical approach: rather than pruning visual tokens, it eliminates the visual KV cache from the drafter entirely via text-anchored window attention (VATA), confining attention strictly to text positions whose hidden states already encode internalized visual semantics from the target model.

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 [18] reuses the target's KV cache directly through learned projections, compressing the multimodal KV before cross-attention to make the speculative module tractable. FastVLM [43] shares the backbone between drafter and verifier, maintaining a single KV cache for both stages and eliminating redundant KV computation. SpecVLM [17] optimizes KV management at the prefill stage, reducing the memory footprint of visual token caching. STAR [20] treats KV-affecting design choices (visual token count, attention head count, interaction layer position) as NAS [21] search dimensions, explicitly optimizing KV scale and memory access patterns. STD [44] exploits attention sparsity by selecting only the top-K

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

KV entries, reducing I/O cost while sharing all parameters with the dense model so no additional GPU memory is needed.

3.3.3. Target-Informed Transfer

These methods tighten draft–target alignment by exposing the drafter to signals derived from the target model, reducing mismatch during speculative generation.

DREAM [19] selects target intermediate layers using attention entropy and injects features via cross-attention at each draft step. STAR [20] identifies optimal distillation layers (high attention concentration, low cross-layer variation) and injects target features through the NAS-searched interaction architecture. SPARROW [37] 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.

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 (§3.3.3), 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 required.

Architectural Inheritance.

SSD [39] 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 [43] 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.

MASSV [22] 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 [57] applies online logit distillation during drafter training, aligning the drafter's output distribution with the target's at each token position.

Representation Alignment.

VISPEC [13] trains a Q-Former-style [14] vision adaptor to produce compressed visual tokens and a global visual vector aligned with the target's visual processing. HIVIS [16] 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 [15] 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, the inference-time Target-Informed Transfer (§3.3.3). For example, SPARROW [37] 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

impacting the overall speedup. We organize verification into execution strategies (§4.1) and verification optimization (§4.2).

4.1. Verification Execution

As shown in Figure 6 and Table 3, this section summarizes various verification criteria, 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).

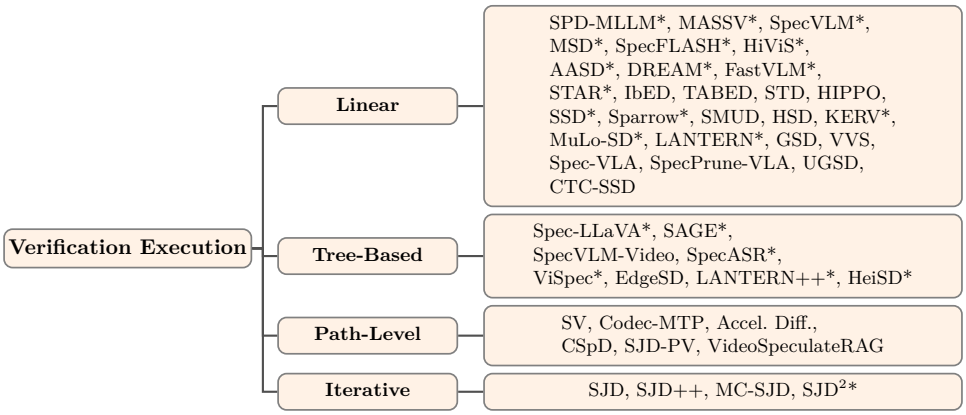

Figure 6. Sub-taxonomy of verification execution strategies (§4.1). Linear verification remains dominant; tree, path, and iterative approaches address domain-specific needs. * denotes methods requiring training/fine-tuning.

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 (Equation 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.

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

4.1.1. Linear Verification (Standard)

Standard verification evaluates K draft tokens left to right. The first token whose acceptance probability (Equation (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 VLM and VideoLM speculative decoding. The following VLM methods all employ standard linear verification without modifying the verification algorithm itself: SPD-MLLM [12], MASSV [22], SpecVLM [17], MSD [25], SpecFLASH [15], HiViS [16], AASD [18], DREAM [19], FastVLM [43], STAR [20], IbED [23], and TABED [24]. LANTERN [28], GSD [30], and VVS [31] employ sequential left-to-right verification

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

for visual token generation, though LANTERN and GSD relax the acceptance criterion from exact matching to latent-space neighborhood or grouped acceptance (§4.2.2), and VVS dynamically skips verification steps for high-confidence tokens. In the VideoLM domain, STD [44], HIPPO [50], and SPARROW [37] retain standard Leviathan-style accept/reject rules. In the speech domain, SSD [39] employs linear verification along a single draft sequence, introducing speech-specific modifications to the acceptance rule (§4.2.2). UGSD [42] also performs sequential left-to-right verification, but combines it with a relaxed rank-based acceptance criterion and uncertainty-gated cloud offloading (§4.2.2, §4.2.1). CTC-SSD [52] employs linear verification of the CTC draft hypothesis through a single LLM forward pass, accepting the hypothesis if all token likelihoods exceed a threshold (§4.2.2); otherwise it falls back to AR decoding from the longest accepted prefix. Rather than standard token exact-matching against a continuous draft, SMUD [41] 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 (§4.2.2).

Linear verification remains prevalent because most multimodal speculative decoding innovations occur during the *draft* phase, improving candidate quality or decreasing drafting cost, rather than the *verification* phase.

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.

SPEC-LLAVA [57] is the primary VLM 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 [13] also uses a tree-based speculative mechanism during generation. SAGE [59] also operates on tree structures, with its entropy-guided shaping determining the tree topology that the target verifies. EDGESD [26] 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.

LANTERN++ [29] extends this 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 (§4.2.2).

In the VideoLM domain, SPECVLM (Video) [60] adopts EAGLE-style [61] static tree structures with tree attention masks, verifying multi-branch candidate trees generated from pruned video tokens.

In the speech domain, SPECASR [40] 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.

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 VLA domain, HEISD [36] 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 (§4.2.2) with adaptive verify-skip (§4.2.1) to select the longest acceptable action sequence.

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.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

SV (Speculative Verdict) [58] performs trajectory-level validation for VLM 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.

VIDEOSPECULATERAG [38] applies candidate-level verification to video RAG, verifying full answer candidates rather than token prefixes. Each retrieved document produces an independent candidate answer via a lightweight draft VLM; the verifier then performs two-stage candidate reranking. In the first stage, the verifier computes a reliability score from the probability that its first generated token is "Yes" given the candidate, its extracted entity, and reasoning trace. Candidates within a tolerance margin δ of the maximum reliability score form a high-confidence set A_H. In the second stage, a CLIP-based entity alignment score reranks candidates in A_H by visual similarity between the extracted entity and video keyframes, selecting the final answer. This candidate-level reranking with tolerance-based filtering is better viewed as path-level selection than classical token-level speculative acceptance.

In codec-based speech synthesis, CODEC-MTP [51] similarly performs sequence-level selection. Rather than accepting or rejecting individual tokens, CODEC-MTP [51] 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 diffusion models, path-level verification evaluates whether a proposed denoising trajectory segment constitutes a valid draw from the target diffusion process, requiring coupling-based acceptance criteria [53].

CSPD [32] enforces density-level consistency for continuous-valued visual AR models through acceptance-rejection sampling over the draft–target density ratio, detailed in §4.2.2.

SJD-PV [48] (Speculative Jacobi Decoding with Phrase Verification) elevates verification granularity from individual tokens to *phrase-levels*. 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 (§4.1.4).

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 [45–47,56] 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 [47] strengthens convergence detection by applying maximal coupling between iterations, maximizing the probability that consecutive steps sample identical tokens. SJD^2 [56] 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

As with draft generation, several verification-side optimization strategies recur across modalities, as shown in Figure 7.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

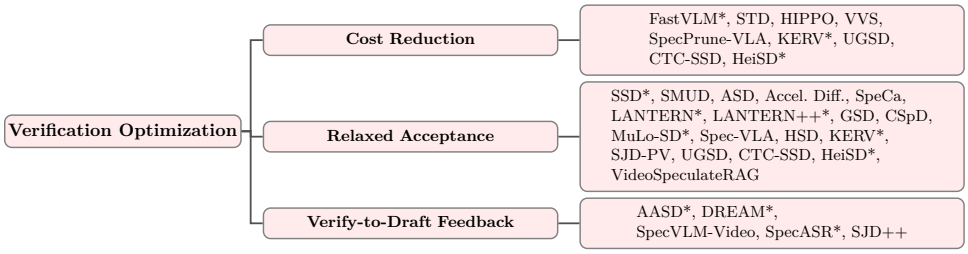

Figure 7. Sub-taxonomy of verification optimization strategies (§4.2). These strategies recur across modalities. * denotes methods requiring training/fine-tuning.

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.

Shared-Backbone KV Reuse.

Methods that share the backbone between draft and verification stages eliminate redundant computation. FastVLM [43] 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 [44] similarly reassigns computational responsibility: the dense 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.

Confidence-Based Verification Skipping.

VVS [31] reduces verification cost by *selectively skipping target model verification* for high-confidence draft tokens, 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 verification skipping parallels SpeCa [55]'s forecast gating (§4.2.2), where accurate cached predictions bypass full recomputation, but operates at the token level rather than the feature level. UGSD [42] 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 [52] 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 [36] 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.

Input Pruning for Shared-Backbone Verification.

SpecPrune-VLA [49] implicitly reduces verification cost through its action-aware pruning framework (§3.3.1): by reducing the visual token count entering the shared backbone, both draft and verification forward passes become cheaper. The self-speculative design, where the pruned model serves as its own drafter, eliminates the need for a separate draft model, a practical advantage for embodied deployment where memory is constrained.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

4.2.2. Relaxed Acceptance

Standard speculative decoding enforces exact distribution matching. Multiple modalities, including T2I, VLA, speech, and diffusion, benefit from relaxing this requirement when perceptual or functional equivalence suffices.

Latent-Space and Continuous-Density Acceptance.

In visual generation, exact token matching yields impractically low acceptance rates due to codebook redundancy. LANTERN [28] and LANTERN++ [29] 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 [30] adopts grouped acceptance, treating visually equivalent token clusters as interchangeable. CSPD [32] extends relaxation to continuous-valued token spaces through density-ratio acceptance-rejection sampling, enabling theoretically grounded verification in continuous output spaces. MULO-SD [33] 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.

Perceptual and Functional Tolerance.

Domain-specific equivalence notions frequently replace strict token matching across modalities. SPEC-VLA [34] accepts draft action tokens whose continuous control signals fall within a task-dependent distance tolerance of the target action. KERV [35] 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 detailed in §3.1.1. HEISD [36] 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.

VIDEOSPECULATERAG [38] 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.

HSD [27] introduces τ-matching for the document parsing domain: 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.

SSD [39] 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 [42] 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 [52] 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. All six method families (VLA tolerance, kinematic rectification, sequence-wise grouping, speech perceptual tolerance, rank-based acceptance, and token-likelihood gating) recognize that multiple distinct tokens can function equivalently when evaluated through a modality-appropriate perceptual or task-specific metric.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

Coupling-Based and Exchangeability-Based Acceptance.

For diffusion models, verification operates over continuous trajectories rather than discrete tokens. ACCELERATED DIFFUSION SAMPLING [53] 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 [54] 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.

SPECA [55] 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. This gating mechanism creates adaptive verification granularity: accurate forecasts pass through cheaply, while degraded forecasts trigger full recomputation.

Dual-Hypothesis Boundary Detection.

Unlike traditional speculative decoding where the verifier merely checks if a proposed token is correct, SMUD [41] 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 [41] 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.

The target-informed transfer mechanisms described in §3.3.3, including AASD [18]'s T-D Attention and DREAM [19]'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.

Verifier Attention as Pruning Signal.

SPECVLM (Video) [60] uses verifier attention scores as a feedback signal to guide draft-side 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.

Rather than treating rejected tokens as wasted computation, several methods across modalities convert failed draft outputs into useful partial results, making recycling a verification-triggered feedback mechanism. SPECASR [40] (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. SPECASR also employs a two-pass sparse-tree structure that branches only at high-uncertainty positions, selectively expanding the candidate space where linear verification is most likely to reject. SJD++ [46] (T2I)

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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 have implemented speculative decoding support. Table 4 summarizes the major frameworks, their speculative decoding capabilities, and their current level of multimodal support.

vLLM.

vLLM [62] 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, EAGLE-3, MTP, 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 [63] 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 and EAGLE-3, and has been actively expanding multimodal model support 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 [64] is NVIDIA's datacenter-oriented inference optimization framework built on a PyTorch-native architecture, 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, 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 [65] 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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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.

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 [62]	EAGLE, EAGLE-3, MTP, Draft Model, n-gram, Suffix	✓	△[a]
SGLang [63]	EAGLE, EAGLE-3, MTP, Standalone Draft, n-gram	✓	×
TensorRT-LLM [64]	EAGLE (1/2/3), Medusa, ReDrafter, MTP, Lookahead, Draft Model, n-gram	✓	×
TensorRT Edge-LLM [65]	EAGLE-3	✓	✓[b]
LMDeploy [66]	Medusa (TreeMask)	✓	×
HF Transformers [67]	Draft Model (`assisted_generation`)	✓	×[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_generation` API itself lacks multimodal-specific optimizations.

LMDeploy.

LMDeploy [66] provides high-performance inference through its TurboMind C++ backend and PyTorch backend, featuring continuous batching and efficient CUDA kernels. It implements speculative decoding via Medusa-style TreeMask verification. Although LMDeploy supports multimodal model deployment (VLMs), its speculative decoding capabilities remain text-focused and are still marked as experimental.

Hugging Face Transformers.

The Hugging Face Transformers library [67] provides a widely-adopted `assisted_generation` API that implements basic speculative decoding with draft models. Among the surveyed methods, HIPPO [50] explicitly builds on Transformers v4.57.0, and DREAM [19] 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.

Beyond production serving frameworks, several open-source algorithm libraries provide reusable implementations of speculative decoding techniques. EAGLE [61] and its successor EAGLE-2 offer feature-level auto-regression heads with tree-structured verification, directly adopted by LANTERN [28], SpecVLM [17], SAGE [59], and DREAM [19]. Medusa [68] provides multi-head parallel drafting, used as a baseline by DREAM [19]. Spec-Bench [3] 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.

Discussion.

A narrowing but persistent gap remains between speculative decoding and multimodal model serving in production inference frameworks. While all major frameworks now support both capabilities

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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).

6. Comparison and Benchmarking

This section provides a systematic comparison of representative multimodal speculative decoding methods. Gathering methods across all domains into a unified benchmarking framework is challenging due to heterogeneous evaluation protocols and base models. However, examining them through our taxonomy reveals consistent architectural trends.

Table 5 highlights four fundamental differences from text-only speculative decoding:

First, *Independent Drafting* dominates Vision–Language and Video–Language Models, mirroring text-only practice. However, multimodal drafters increasingly inject target-model features to improve speculation accuracy (DREAM [19], STAR [20], AASD [18]). Conversely, domains with strong self-drafting structures, such as video KV-splits (STD [44]), layer-sharing (FastVLM [43]), and Jacobi iteration (SJD [45] family), adopt *Shared Backbone* mechanisms. Speech methods similarly span Independent (e.g., SSD [39]) and Shared Backbone (e.g., Codec-MTP [51]) paradigms. Diffusion models uniquely introduce *Drafter-Free Speculation* approaches that generate trajectory segments rather than discrete tokens.

Second, Text-to-Image (T2I) generation reveals two distinct paradigms. Methods like LANTERN [28] and GSD [30] retain the dual-model framework but *relax acceptance criteria* in the visual latent space, which CSpD [32] adapts for continuous-valued representations; MuLo-SD [33] 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 [45] family eliminates separate drafters entirely through *Jacobi self-drafting*, and SJD-PV [48] 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 [47] (3.8–4.2×), EdgeSD [26] (3–5×), and GSD [30] (≈ 3.8×). However, the highest reported VLM speedups still generally require dedicated drafter training or distillation (e.g., Spec-LLaVA [57], STAR [20], DREAM [19]).

Finally, strict match verification dominates discrete-token multimodal generation (VLMs and Video–Language Models). Text-to-Image (T2I), Vision–Language–Action (VLA), Speech/Audio, and Diffusion Transformer (DiT) models more frequently rely on relaxed, convergence-based, or continuous-state criteria to achieve practical viability.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

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 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.

Methods	Drafting			Verification		Representative Target	Speedup (rep.)
	Draft Mechanism	Architecture / Approach	Tuning-free	Criterion	Pattern		
Vision–Language Models							
SPD-MLLM	Independent	Text-Only LM	× (Pretrain)	Strict	Linear	LLaVA (LLaMA)	$\lesssim 2.37\times$
MASSV	Independent	Small VLM	× (Distillation)	Strict	Linear	Qwen2.5-VL, Gemma3	$\lesssim 1.46\times$
ViSpec	Independent	Visual Adaptor	× (Tuning)	Strict	Tree	LLaVA-1.6, Qwen2.5-VL	$\lesssim 3.22\times$
SpecVLM	Independent	Visual Compressor	× (Tuning)	Strict	Linear	LLaVA, Qwen2-VL	$\lesssim 2.9\times$
FastVLM	Shared Backbone	Early Exiting	× (Imitation)	Strict	Linear	LLaVA-1.5	$1.55\times \sim 1.85\times$
MSD	Independent	Text-Vision Decouple	✓	Strict	Linear	LLaVA-1.5	$\lesssim 2.46\times$
SpecFLASH	Independent	Semi-AR Heads	× (Tuning)	Strict	Linear	LLaVA, Qwen-VL	$\lesssim 2.55\times$
Spec-LLaVA	Independent	Token Tree	× (Tuning)	Strict	Tree	LLaVA-1.5	$\lesssim 3.28\times$
SAGE	Independent	Adaptive Tree	× (Tuning)	Strict	Tree	LLaVA-OV, Qwen2.5-VL	$\lesssim 3.36\times$
HiViS	Independent	Visual Token Hiding	× (Tuning)	Strict	Linear	Qwen2.5-VL	$\lesssim 3.15\times$
AASD	Independent	T-D Attention	× (T-D Attn)	Strict	Linear	LLaVA	$\lesssim 2.0\times$
DREAM	Independent	Target-Informed	× (Tuning)	Strict	Linear	LLaVA-v1.6	$\lesssim 3.6\times$
STAR	Independent	NAS + Target Feat.	× (OFA Tuning)	Strict	Linear	LLaVA-v1.6	$\lesssim 3.8\times$
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	$\lesssim 1.74\times$
EdgeSD	Independent	VED + ITM + Tree	✓	Strict	Tree	LLaVA-OV, InternVL2.5	$3.04\times \sim 5.12\times$
SV (Verdict)*	Independent	Small VLM	✓	Path-Level (NLL)	Path	Qwen2.5-VL	N/A
HSD‡	Independent	Pipeline Draft	✓	Relaxed (τ-Tol.)	Hierarchical	Qwen3-VL	$\lesssim 4.89\times$
Text-to-Image Models							
LANTERN	Independent	Small AR Model	× (Tuning)	Relaxed (Latent)	Linear	LlamaGen	$1.75\times \sim 1.82\times$
LANTERN++	Independent	Static Tree	× (Tuning)	Relaxed (Latent)	Tree	LlamaGen	$\approx 2.56\times$
GSD	Independent	Dynamic Clustering	✓	Relaxed (Grouped)	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	× (Fine-tune)	Convergence	Iterative	LlamaGen	$\approx 4.0\times$
MC-SJD	Shared Backbone	Gumbel Coupling	✓	Convergence	Iterative	LlamaGen	$3.8\times \sim 4.2\times$
VVS	Independent	Conf. Skip	✓	Conf. Skip	Linear	LlamaGen	$\approx 2.8\times$
SJD++	Shared Backbone	Token Reuse	× (Fine-tune)	Convergence	Iterative	LlamaGen	$\approx 2.4\times$
CSpD	Independent	Density-Ratio Sampling	✓	Path-Level (Density)	Path	MAR	$\lesssim 2.33\times$
SJD-PV	Shared Backbone	Phrase Library + Jacobi	✓	Phrase-Level (Joint)	Path	Lumina-mGPT	$\lesssim 2.71\times$
MuLo-SD	Independent	Multi-Scale Drafting	× (Training)	Relaxed Local (Neighborhood)	Linear	Tar-1.5B	$\lesssim 1.7\times$
Vision–Language–Action Models							
Spec-VLA	Independent	Small VLA	✓	Relaxed (Action)	Linear	OpenVLA	$\approx 1.42\times$
SpecPrune-VLA	Shared Backbone	Action-Aware Pruning	✓	Strict	Linear	OpenVLA-OFT, π_0	$1.46\times \sim 1.57\times$
KERV	Independent	KF-Rectified Draft	× (Draft Train)	Relaxed (Kinematic)	Linear	OpenVLA (LIBERO)	$1.48\times \sim 1.57\times$
HeiSD	Independent	Hybrid Retrieval + Drafter	× (Draft Train)	Relaxed (Seq-Wise)	Tree	OpenVLA (LIBERO)	$\lesssim 2.45\times$
Video–Language Models							
STD	Shared Backbone	Sparse KV Routing	✓	Strict	Linear	Qwen2-VL, LLaVA-OV	$\lesssim 1.94\times$
SpecVLM (Video)	Independent	Token Tree	✓	Strict	Tree	LLaVA-OV	$\lesssim 2.68\times$
HIPPO	Shared Backbone	Pipeline Overlap	✓	Strict	Linear	LLaVA-OV	$\lesssim 3.51\times$
Sparrow	Independent	HSR + VATA	× (Training)	Strict	Linear	LLaVA-OV, Qwen2.5-VL	$\lesssim 2.82\times$
VideoSpeculateRAG	Independent	Small VLM + Per-Doc Parallel	✓	Tolerance-Based (δ)	Path	Qwen2.5-VL-32B	$\approx 2\times$
Speech and Audio Models							
SSD	Independent	Small Audio LM	× (Distillation)	Relaxed (β-Tol.)	Linear	CosyVoice-2	$\approx 1.4\times$
SpecASR	Independent	Adaptive Len. + Tree	× (Tuning)	Strict (w/ Repair)	Tree	Llama, Vicuna	$3.04\times \sim 3.79\times$
SMUD	Independent	CTC Pseudo-Draft	✓	Dual-Hypothesis	Linear	E-Branchformer ASR	$\approx 1.4\times$
Codec-MTP	Shared Backbone	Block Prediction	✓	Path-Level (Viterbi)	Path	VALL-E / USLM	$4.0\times \sim 5.0\times$
UGSD	Independent	Edge–Cloud Uncertainty Gate	✓	Relaxed (Top-R Rank)	Linear	Qwen2.5-Omni-3B / Qwen3-Omni	1.40× latency
CTC-SSD	Shared Backbone	CTC Encoder Draft	✓	Relaxed (Likelihood τ)	Linear	Granite-Speech-1B	$\lesssim 4.4\times$
Diffusion Models							
SpeCa	Drafter-Free Speculation	Feature Caching	✓	Relaxed (Feat.)	Path	DiT, FLUX, HunyuanVideo	$\lesssim 7.3\times$
Accel. Diff.	Drafter-Free Speculation	Coupling Jumps	✓	Path-Level (Coupling)	Path	DiT, EDM	$2.0\times \sim 3.0\times$
ASD	Drafter-Free Speculation	Stochastic Exchange	✓	Convergence	Path	DDPM	$1.8\times \sim 4.0\times$

6.1. Unified VLM Benchmarking

While Table 5 collects self-reported speedups under heterogeneous settings, a fair comparison requires controlled evaluation. MMSpec [69] provides the first standardized benchmark for speculative decoding in vision–language models, evaluating six diverse VLM subtasks (General VQA, Text VQA, Image Captioning, Chart QA, Complex Reasoning, and Multi-turn Conversation) across both vision-agnostic methods (EAGLE-1/2/3, Medusa) and vision-aware methods (MSD, ViSpec) on Qwen2.5-VL-7B and LLaVA-1.5-7B.

Three key findings emerge from the MMSpec evaluation. First, vision-agnostic methods designed for text-only LLMs degrade sharply on VLMs. EAGLE-3 drops below the autoregressive baseline on both models (0.96× on Qwen2.5-VL-7B, 0.70× on LLaVA-1.5-7B), meaning speculation actually slows down inference, because these methods ignore the visual-conditioned token distribution. Second, vision-aware methods consistently outperform all vision-agnostic baselines. MSD achieves 2.58× overall speedup on Qwen2.5-VL-7B, and ViSpec reaches 2.58× on LLaVA-1.5-7B, confirming that modeling the interaction between visual inputs and language generation is necessary for effective multimodal speculative decoding. Third, performance varies substantially across subtasks: MSD on Qwen2.5-VL-7B ranges from 1.80× (Text VQA) to 4.06× (Complex Reasoning), reflecting differences in output length and visual dependency. This instability highlights the need for adaptive speculation

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

strategies. MMSpec also reports that vision awareness becomes critical at larger batch sizes and that throughput speedup does not reliably reflect latency performance [69].

The absence of similar standardized benchmarks for Text-to-Image, VLA, Video, Speech, and Diffusion domains (§7) makes direct comparison across these modalities difficult, and the speedup figures in Table 5 remain incomparable as 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 multiformat 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. Speculative decoding is fundamentally a memory-bound optimization technique; however, as the KV cache grows linearly with sequence length and attention memory-access cost scales increasingly with longer contexts, memory transactions overshadow arithmetic operations. 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 discrete token generation (VLMs and Audio LMs), 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, is a critical open problem in real-time multimodal deployment.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

Cross-Pollinating Representation-Level Verification to VLMs.

Current VLM speculative decoding relies on strict token matching, forcing the drafter to predict the target's exact discrete token sequence. However, as demonstrated by several Text-to-Image (T2I) methods (§4.2.2), relaxing verification to the latent or representation space substantially improves acceptance rates. A major future direction lies in adapting 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, could break the current $\approx 2\times$ speedup ceiling of tuning-free VLM speculation without requiring costly self-correction.

Eliminating Drafters via Jacobi Self-Drafting in VideoLMs.

A major opportunity for Video–Language Models is the elimination of the external drafter entirely. Inspired by recent breakthroughs in Jacobi-style self-drafting for T2I generation (§4.1.4), applying *Jacobi-accelerated self-drafting* directly to the parallelizable temporal attention heads of a VideoLM could yield substantial speedups. Because consecutive video frames exhibit high temporal redundancy, a VideoLM could initialize a draft trajectory by copying the prior frame's KV activations and simultaneously verifying across the temporal dimension in a single forward pass. This constraint manipulation, dropping the requirement that tokens must be generated strictly autoregressively, could enable tuning-free, unbounded speculation horizons for long-form video understanding.

Preserving Temporal Coherence in Video Generation.

Speculative decoding for video generation faces the challenge of maintaining spatio-temporal consistency across long horizons. Unlike text, where an incorrect token at one position may not ruin the entire sequence, a mispredicted frame deep in the draft can trigger cascading verification failures and corrupt the object permanence or motion coherence of subsequent frames. Developing temporally-aware verification metrics, such as optical-flow-guided acceptance criteria or object-centric latency bounds, could help verify multi-frame drafts more robustly.

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 [69] have introduced standardized benchmarks for vision–language models (§6.1), no equivalent evaluation platform exists for Text-to-Image generation, Vision–Language–Action control, Video–Language understanding, Speech/Audio models, 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, ones that isolate algorithmic efficiency from system-level engineering and define modality-appropriate quality metrics, is essential for transparent and reproducible progress.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

8. Conclusions

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), Text-to-Image (T2I) systems, Vision–Language–Action (VLA) agents, Video–Language models, Speech systems, 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, 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 bounds in robotic control, and tolerance for stochastic variance in diffusion processes. As foundation models increasingly adopt native multimodal generation, autoregressive latency will become a critical deployment constraint. Speculative decoding provides an effective pathway to achieve 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

1. Leviathan, Y.; Kalman, M.; Matias, Y. Fast inference from transformers via speculative decoding. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 19274–19286.
2. Chen, C.; Borgeaud, S.; Irving, G.; Lespiau, J.B.; Sifre, L.; Jumper, J. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318* **2023**.
3. Xia, H.; Yang, Z.; Dong, Q.; Wang, P.; Li, Y.; Ge, T.; Liu, T.; Li, W.; Sui, Z. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *Findings of the Association for Computational Linguistics: ACL 2024* **2024**, pp. 7655–7671.
4. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. *Advances in neural information processing systems* **2023**, *36*, 34892–34916.
5. Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; Zhou, J. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond* **2023**, *2*, 1.
6. Esser, P.; Rombach, R.; Ommer, B. Taming transformers for high-resolution image synthesis. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 12873–12883.
7. Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Proceedings of the Conference on Robot Learning. PMLR, 2023, pp. 2165–2183.
8. Lin, B.; Ye, Y.; Zhu, B.; Cui, J.; Ning, M.; Jin, P.; Yuan, L. Video-llava: Learning united visual representation by alignment before projection. In Proceedings of the Proceedings of the 2024 conference on empirical methods in natural language processing, 2024, pp. 5971–5984.
9. Défossez, A.; Copet, J.; Synnaeve, G.; Adi, Y. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438* **2022**.
10. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems* **2020**, *33*, 6840–6851.
11. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* **2020**.
12. Gagrani, M.; Goel, R.; Jeon, W.; Park, J.; Lee, M.; Lott, C. On speculative decoding for multimodal large language models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 8285–8289.
13. Kang, J.; Shu, H.; Li, W.s.; Zhai, Y.; Chen, X. ViSpec: Accelerating vision-language models with vision-aware speculative decoding. *arXiv preprint arXiv:2509.15235* **2025**.
14. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Proceedings of the International conference on machine learning. PMLR, 2023, pp. 19730–19742.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

15. Wang, Z.; Li, R.; Du, H.; Zhou, J.T.; Zhang, Y.; Yang, X. FLASH: Latent-Aware Semi-Autoregressive Speculative Decoding for Multimodal Tasks. *arXiv preprint arXiv:2505.12728* **2025**.
16. Xie, Z.; Wang, P.; Qiu, S.; Cheng, J. HiViS: Hiding Visual Tokens from the Drafter for Speculative Decoding in Vision-Language Models. *arXiv preprint arXiv:2509.23928* **2025**.
17. Huang, H.; Yang, F.; Liu, Z.; Yin, X.; Li, D.; Ren, P.; Barsoum, E. SpecVLM: Fast Speculative Decoding in Vision-Language Models. *arXiv preprint arXiv:2509.11815* **2025**.
18. Yang, C.; Chen, R.; Zhang, M.; Pang, W.; Chen, Y.; Xu, R.; Fu, K.; Wang, C.; Gao, L. AASD: Accelerate Inference by Aligning Speculative Decoding in Multimodal Large Language Models. In Proceedings of the 2025 62nd ACM/IEEE Design Automation Conference (DAC). IEEE, 2025, pp. 1–7.
19. Hu, Y.; Xia, T.; Liu, Z.; Raman, R.; Liu, X.; Bao, B.; Sather, E.; Thangarasa, V.; Zhang, S.Q. Dream: Drafting with refined target features and entropy-adaptive cross-attention fusion for multimodal speculative decoding. *arXiv preprint arXiv:2505.19201* **2025**.
20. Liu, Z.; Hu, Y.; Xia, T.; Bao, B.; Sather, E.; Thangarasa, V.; Zhang, S.Q. STAR: Speculative Decoding with Searchable Drafting and Target-Aware Refinement for Multimodal Generation. *OpenReview preprint* **2025**.
21. Cai, H.; Gan, C.; Wang, T.; Zhang, Z.; Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791* **2019**.
22. Ganesan, M.; Segal, S.; Aggarwal, A.; Sinnadurai, N.; Lie, S.; Thangarasa, V. MASSV: Multimodal adaptation and self-data distillation for speculative decoding of vision-language models. *arXiv preprint arXiv:2505.10526* **2025**.
23. Lee, M.; Kang, W.; Ahn, B.; Classen, C.; Yan, M.; Koo, H.I.; Lee, K. In-batch Ensemble Drafting: Robust Speculative Decoding for LVLMs. In Proceedings of the First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models, 2025.
24. Lee, M.; Kang, W.; Ahn, B.; Classen, C.; Galim, K.; Oh, S.; Yan, M.; Koo, H.I.; Lee, K. TABED: Test-Time Adaptive Ensemble Drafting for Robust Speculative Decoding in LVLMs. *arXiv preprint arXiv:2601.20357* **2026**.
25. Lin, L.; Lin, Z.; Zeng, Z.; Ji, R. Speculative decoding reimagined for multimodal large language models. *arXiv preprint arXiv:2505.14260* **2025**.
26. Huang, H.; Zhan, W.; Duan, H.; Peng, K.; Min, G.; Zhao, Z.; Zhao, Z.; Ye, Y. EdgeSD: Efficient Speculative Decoding with Vision-Decoding Disaggregation for MLLM Inference in Edge-Cloud Networks. *IEEE Transactions on Mobile Computing* **2026**.
27. Liao, W.; Li, H.; Xie, P.; Cai, X.; Shen, Y.; Xin, Y.; Qin, Q.; Ye, S.; Li, T.; Hu, M.; et al. Training-Free Acceleration for Document Parsing Vision-Language Model with Hierarchical Speculative Decoding. *arXiv preprint arXiv:2602.12957* **2026**.
28. Jang, D.; Park, S.; Yang, J.Y.; Jung, Y.; Yun, J.; Kundu, S.; Kim, S.Y.; Yang, E. LANTERN: Accelerating Visual Autoregressive Models with Relaxed Speculative Decoding. *arXiv preprint arXiv:2410.03355* **2025**. ICLR 2025.
29. Jang, D.; Park, S.; Yang, J.Y.; Jung, Y.; Yun, J.; Kundu, S.; Kim, S.Y.; Yang, E. LANTERN++: Enhanced Relaxed Speculative Decoding with Static Tree Drafting for Visual Auto-regressive Models. *arXiv preprint arXiv:2502.06352* **2025**. ICLR 2025 SCOPE Workshop.
30. So, J.; Shin, J.; Kook, H.; Park, E. Grouped Speculative Decoding for Autoregressive Image Generation. *arXiv preprint arXiv:2508.07747* **2025**. ICCV 2025.
31. Dong, H.; Li, Y.; Lu, R.; Tang, C.; Xia, S.T.; Wang, Z. VVS: Accelerating Speculative Decoding for Visual Autoregressive Generation via Partial Verification Skipping. *arXiv preprint arXiv:2511.13587* **2025**.
32. Wang, Z.; Zhang, R.; Ding, K.; Yang, Q.; Li, F.; Xiang, S. Continuous speculative decoding for autoregressive image generation. *arXiv preprint arXiv:2411.11925* **2024**.
33. Peruzzo, E.; Sautière, G.; Habibian, A. Multi-Scale Local Speculative Decoding for Image Generation. *arXiv preprint arXiv:2601.05149* **2026**.
34. Wang, S.; Yu, R.; Yuan, Z.; Yu, C.; Gao, F.; Wang, Y.; Wong, D.F. Spec-VLA: Speculative Decoding for Vision-Language-Action Models with Relaxed Acceptance. *arXiv preprint arXiv:2507.22424* **2025**.
35. Zheng, Z.; Mao, Z.; Li, M.; Chen, J.; Sun, X.; Zhang, Z.; Cao, D.; Mei, H.; Chen, X. KERV: Kinematic-Rectified Speculative Decoding for Embodied VLA Models. In Proceedings of the Proceedings of the 63rd ACM/IEEE Design Automation Conference (DAC), 2026.
36. Zheng, Z.; Mao, Z.; Tian, S.; Li, M.; Chen, J.; Sun, X.; Zhang, Z.; Liu, X.; Cao, D.; Mei, H.; et al. HeiSD: Hybrid Speculative Decoding for Embodied Vision-Language-Action Models with Kinematic Awareness. *arXiv preprint arXiv:2603.17573* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

37. Zhang, L.; Zhang, Z.; Hong, W.; Qiao, P.; Li, D. Sparrow: Text-Anchored Window Attention with Visual-Semantic Glimpsing for Speculative Decoding in Video LLMs. *arXiv preprint arXiv:2602.15318* **2026**.
38. Li, G.; Liu, P. FastV-RAG: Towards Fast and Fine-Grained Video QA with Retrieval-Augmented Generation. *arXiv preprint arXiv:2601.01513* **2026**.
39. Lin, Z.; Zhang, Y.; Yuan, Y.; Yan, Y.; Liu, J.; Wu, Z.; Hu, P.; Yu, Q. Accelerating Autoregressive Speech Synthesis Inference With Speech Speculative Decoding. *arXiv preprint arXiv:2505.15380* **2025**.
40. Wei, L.; Zhong, S.; Xu, S.; Wang, R.; Huang, R.; Li, M. SpecASR: Accelerating LLM-based Automatic Speech Recognition via Speculative Decoding. In Proceedings of the 2025 62nd ACM/IEEE Design Automation Conference (DAC). IEEE, 2025, pp. 1–7.
41. Okabe, K.; Yamamoto, H. Simultaneous Masked and Unmasked Decoding with Speculative Decoding Masking for Fast ASR without Accuracy Loss. In Proceedings of the Proc. Interspeech 2025, 2025, pp. 634–638.
42. Xue, X.; Lu, J.; Gao, Y.; Huang, G.; Dang, T.; Jia, H. Edge–Cloud Collaborative Speech Emotion Captioning via Token-Level Speculative Decoding in Audio-Language Models. *arXiv preprint arXiv:2603.11397* **2026**.
43. Bajpai, D.J.; Hanawal, M.K. FastVLM: Self-Speculative Decoding for Fast Vision-Language Model Inference. In Proceedings of the 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, 2025, pp. 1166–1183.
44. Zhang, X.; Du, C.; Yu, S.; Wu, J.; Zhang, F.; Gao, W.; Liu, Q. Sparse-to-Dense: A Free Lunch for Lossless Acceleration of Video Understanding in LLMs. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2025, pp. 734–742.
45. Teng, Y.; Shi, H.; Liu, X.; Ning, X.; Dai, G.; Wang, Y.; Li, Z.; Liu, X. Accelerating Auto-regressive Text-to-Image Generation with Training-free Speculative Jacobi Decoding. *arXiv preprint arXiv:2410.01699* **2025**. ICLR 2025.
46. Teng, Y.; Jiang, Z.; Shi, H.; Liu, X.; Ning, X.; Dai, G.; Wang, Y.; Li, Z.; Liu, X. SJD++: Improved Speculative Jacobi Decoding for Training-free Acceleration of Discrete Auto-regressive Text-to-Image Generation. *arXiv preprint arXiv:2512.07503* **2025**.
47. So, J.; Kook, H.; Jang, C.; Park, E. MC-SJD: Maximal Coupling Speculative Jacobi Decoding for Autoregressive Visual Generation Acceleration. *arXiv preprint arXiv:2510.24211* **2025**.
48. Yu, Z.; Zhang, B.; Shan, B.; Liu, X.; Zhou, D.; Liang, G.; Ye, G.; Ye, Y. SJD-PV: Speculative Jacobi Decoding with Phrase Verification for Autoregressive Image Generation. *arXiv preprint arXiv:2603.06666* **2026**.
49. Wang, H.; Xu, J.; Pan, J.; Zhou, Y.; Dai, G. SpecPrune-VLA: Accelerating Vision-Language-Action Models via Action-Aware Self-Speculative Pruning. *arXiv preprint arXiv:2509.04043* **2025**.
50. Lv, Q.; Liu, T.; Wu, W.; Xu, X.; Zhou, B.; Wu, F.; Zhang, C. HIPPO: Accelerating Video Large Language Models Inference via Holistic-aware Parallel Speculative Decoding. *arXiv preprint arXiv:2601.08273* **2026**.
51. Nguyen, T.D.; Kim, J.H.; Choi, J.; Choi, S.; Park, J.; Lee, Y.; Chung, J.S. Accelerating codec-based speech synthesis with multi-token prediction and speculative decoding. In Proceedings of the ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025, pp. 1–5.
52. Saon, G.; Thomas, S.; Fukuda, T.; Nagano, T.; Dekel, A.; Lastras, L. Self-Speculative Decoding for LLM-based ASR with CTC Encoder Drafts. *arXiv preprint arXiv:2603.11243* **2026**.
53. De Bortoli, V.; Galashov, A.; Gretton, A.; Doucet, A. Accelerated diffusion models via speculative sampling. *arXiv preprint arXiv:2501.05370* **2025**.
54. Hu, H.; Das, A.; Sadigh, D.; Anari, N. Diffusion Models are Secretly Exchangeable: Parallelizing DDPMs via Autospeculation. *arXiv preprint arXiv:2505.03983* **2025**.
55. Liu, J.; Zou, C.; Lyu, Y.; Ren, F.; Wang, S.; Li, K.; Zhang, L. Speca: Accelerating diffusion transformers with speculative feature caching. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, 2025, pp. 10024–10033.
56. Teng, Y.; Wang, F.; Liu, X.; Chen, Z.; Shi, H.; Wang, Y.; Li, Z.; Liu, W.; Zou, D.; Liu, X. Speculative Jacobi-Denoising Decoding for Accelerating Autoregressive Text-to-image Generation. *arXiv preprint arXiv:2510.08994* **2025**. NeurIPS 2025.
57. Huo, M.; Zhang, J.; Wang, H.; Xu, J.; Chen, Z.; Tai, H.; Chen, Y. Spec-LLaVA: Accelerating Vision-Language Models with Dynamic Tree-Based Speculative Decoding. *arXiv preprint arXiv:2509.11961* **2025**.
58. Liu, Y.; Qin, L.; Wang, S. Small drafts, big verdict: Information-intensive visual reasoning via speculation. *arXiv preprint arXiv:2510.20812* **2025**.
59. Tong, Y.; Zhang, T.; Wan, Y.; Lin, K.; Yuan, J.; Hu, C. SAGE: Accelerating Vision-Language Models via Entropy-Guided Adaptive Speculative Decoding. *arXiv preprint arXiv:2602.00523* **2026**.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 March 2026 doi:10.20944/preprints202603.2344.v1

60. Ji, Y.; Zhang, J.; Xia, H.; Chen, J.; Shou, L.; Chen, G.; Li, H. Specvlm: Enhancing speculative decoding of video llms via verifier-guided token pruning. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 7216–7230.
61. Li, Y.; Wei, F.; Zhang, C.; Zhang, H. EAGLE: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077* **2024**.
62. Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C.H.; Gonzalez, J.; Zhang, H.; Stoica, I. Efficient memory management for large language model serving with pagedattention. In Proceedings of the Proceedings of the 29th symposium on operating systems principles, 2023, pp. 611–626.
63. Zheng, L.; Yin, L.; Xie, Z.; Sun, C.; Huang, J.; Yu, C.H.; Cao, S.; Kozyrakis, C.; Stoica, I.; Gonzalez, J.E.; et al. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems* **2024**, *37*, 62557–62583.
64. NVIDIA. TensorRT-LLM. https://github.com/NVIDIA/TensorRT-LLM, 2024. Accessed: 2026-03-11.
65. 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.
66. 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.
67. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* **2019**.
68. Cai, T.; Li, Y.; Geng, Z.; Peng, H.; Lee, J.D.; Chen, D.; Dao, T. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774* **2024**.
69. Shen, H.; Wang, X.; Zhang, P.; Hsieh, Y.; Han, Q.; Wan, Z.; Zhang, Z.; Zhang, J.; Xiong, J.; Liu, Z.; et al. MMSpec: Benchmarking Speculative Decoding for Vision-Language Models. *arXiv preprint arXiv:2603.14989* **2026**.

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

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.