Latent Diffusion Models: A Survey on Foundations, Variants, and Web-scale Deployments

Jee-Woo Shin[1], Chayapol Kamyod[2] and Chung-Pyo Hong[1,*]

[1]*Division of Computer Engineering, Hoseo University, Republic of Korea*
[2]*Computer and Communication Engineering for Capacity Building Research Center, Mae Fah Luang University, Thailand*
E-mail: jwshin2022@gmail.com; chayapol.kam@mfu.ac.th; cphong@hoseo.edu
[*]*Corresponding Author*

Received 14 December 2025; Accepted 17 January 2026

Abstract

Latent diffusion models (LDMs) have rapidly become the de facto backbone of web-scale generative systems, powering text-to-image platforms such as Stable Diffusion and their video, 3D, and domain-specific extensions. By performing the diffusion process in a compressed latent space rather than directly in pixel space, LDMs achieve a favorable trade-off between computational efficiency and generative fidelity, enabling deployment in interactive web applications and large-scale content pipelines. This paper presents a comprehensive survey of LDMs from the perspective of both foundational modeling and web engineering. We first review the background of diffusion models and latent representations, contrasting LDMs with classical VAEs, GANs, and pixel-space diffusion models. We then dissect the architectural design of LDMs, including autoencoder backbones, latent-space U-Nets and diffusion transformers, conditioning mechanisms, training objectives, and sampling accelerations. Building on recent general surveys of diffusion models in vision, temporal data, and inverse problems, we propose a taxonomy of LDM variants, covering 2D image models, video and 4D models, and

Journal of Web Engineering, Vol. 25_4, 599–634.
doi: 10.13052/jwe1540-9589.2546

domain-specific LDMs in medical imaging, watermarking, time series, and text. From a web engineering viewpoint, we analyze LDM-based services exposed via web APIs, hosted user interfaces, and developer platforms, and discuss system-level concerns such as scalability, latency, cost, safety, and governance. We review current evaluation methodologies (quality, diversity, downstream task performance, robustness, watermarking) and highlight open challenges in controllability, interpretability, resource efficiency, and regulatory compliance, especially in light of recent legal and societal developments around generative deepfakes and copyright. This survey aims to provide both a conceptual map of LDM research and practical guidance for designing, deploying, and governing LDM-driven web systems.

Keywords: Latent diffusion models, diffusion models, generative AI, web engineering, Stable Diffusion, text-to-image, video diffusion, medical imaging, watermarking, content moderation, web services.

1 Introduction

The last few years have seen diffusion models emerge as the dominant paradigm in deep generative modeling, surpassing GANs and VAEs in image quality and diversity across many benchmarks. Comprehensive surveys give broad overviews of diffusion models in vision and beyond, covering theory, variants, and applications; representative examples include the general diffusion survey by [2], the vision-focused survey by [3], and the design-oriented survey by [4]. A more application-centered overview is provided in [18]. However, most of these surveys treat latent diffusion either as one example among many architectures or focus on specific domains (e.g., medical imaging, watermarking, or time series) as discussed in [5, 12–14], and [17].

Latent diffusion models (LDMs) were introduced as a way to push diffusion models to high-resolution image synthesis with tractable training and inference costs by performing the diffusion process in the latent space of a learned autoencoder. The seminal formulation of LDMs appears in [1], which combines a perceptual autoencoder with a latent-space diffusion backbone and cross-attention conditioning. This idea underlies widely used open-source models such as Stable Diffusion and its successors (e.g., SDXL, SD 3.x), which are extensively deployed through web APIs, browser-based interfaces, and SaaS platforms documented in [7] and in the Stability developer documentation [8, 24]. These systems have transformed how web applications generate and manipulate images, videos, and other media, especially in creative, entertainment, and design workflows.

1.1 Motivation and Scope

From a modeling perspective, LDMs occupy a sweet spot: by combining perceptual compression with diffusion in the latent space, they retain the expressive power of diffusion models while significantly reducing memory and computational requirements, as shown in [1] and further discussed in design-centric surveys such as [4]. From a web-engineering perspective, this efficiency is essential to support interactive latency, large-scale content workflows, and cost-effective cloud deployment for web services that expose LDMs over HTTP APIs or interactive web user interfaces, as exemplified by the platforms in [19, 22, 23], and [8, 24].

Existing diffusion surveys are broad but not tailored to (i) the specific design space of LDMs and (ii) their system- and web-oriented implications. The general diffusion survey by [2] provides a taxonomy of diffusion methods and applications but treats the latent versus pixel-space design choice as one axis among many. The vision-oriented survey by [3] focuses on computer-vision tasks and model families, and the design fundamentals survey by [4] emphasizes component-wise design but does not delve deeply into web deployment. Domain-specific surveys in medical imaging [12–14], inverse problems [15], text [16], and time series [17, 20] cover subsets of LDM applications but lack a unified web-engineering lens. The comprehensive survey of diffusion models and their applications in [18] again does not center LDMs as a distinct architectural family.

A leading venue in web engineering explicitly welcomes survey articles that connect technical foundations to concrete web systems, as described in its Aims and Scope and Guidelines for Authors [25]. Our work follows the tradition of earlier surveys on faceted search [26] and question-answering systems [27] by taking a web-centric perspective on a broader technical field. In this spirit, we position this survey at the intersection of latent diffusion model (LDM) research and web-scale system design. We emphasize both the algorithmic foundations of LDMs and the practical aspects of integrating them into web services, APIs, and content pipelines.

1.2 Contributions

This paper makes the following contributions:

- Unified LDM-centric perspective. We synthesize the rapidly growing literature on latent diffusion models, starting from the foundational work in [1], the SDXL architecture in [7], and extensions to video in [6] and platform-level deployments in [8, 24], while relating these to general diffusion surveys [2–4, 18].

- Architectural survey. We provide a component-wise analysis of LDMs, covering autoencoders, latent diffusion backbones, conditioning mechanisms, training objectives, and sampling strategies, drawing primarily on [1, 7], and domain-specific implementations such as [9, 10], and [11].
- Taxonomy of LDM variants. We propose a taxonomy spanning 2D image, video/4D, medical, watermarking, textual, and temporal LDMs, integrating results from domain-specific surveys and application studies in [6, 9–17, 20], and [5].
- Web-engineering angle. We analyze LDM-based services (e.g., Stable Diffusion APIs, web UIs, and hosted spaces) as web systems, discussing latency, scalability, deployment patterns, content pipelines, and safety controls, taking practical examples from [19, 22, 23], and [8, 24].
- Evaluation and governance. We review evaluation methodologies and discuss legal, ethical, and societal issues (e.g., copyright, deepfakes, watermarking) that are particularly salient for web-exposed LDM services, referencing recent legal cases and policy discussions such as those reported in [29] and [28], alongside technical proposals for watermarking and provenance in [5] and broader governance perspectives in [18] and [25].

1.3 Organization

Section 2 reviews background on generative models, diffusion, and latent representations. Section 3 dissects LDM architecture. Section 4 proposes a taxonomy of LDM variants. Section 5 surveys evaluation methodologies. Section 6 addresses system-level considerations and web-scale applications. Section 7 discusses safety, ethics, and governance. Section 8 outlines open research directions, and Section 9 concludes.

2 Background: From Generative Models to Latent Diffusion

2.1 Generative Models in Brief

Before the recent advances in diffusion models, deep generative modeling on the web was dominated by four major families:

- Variational autoencoders (VAEs): Learn a probabilistic encoder and decoder, optimizing an ELBO that balances reconstruction fidelity and latent regularization.

- Generative adversarial networks (GANs): Use a generator–discriminator game to learn realistic samples but these often suffer from mode collapse and training instability.
- Autoregressive models: Factorize likelihoods into sequential conditionals (e.g., PixelCNN, GPT-style language models).
- Normalizing flows: Learn invertible transformations with exact likelihoods but architectural constraints.

Figure 1 provides a compact map of the deep generative modeling landscape and clarifies where latent diffusion models (LDMs) fit within it. Specifically, it highlights diffusion models as a complementary family alongside VAEs, GANs, autoregressive models, and normalizing flows, while

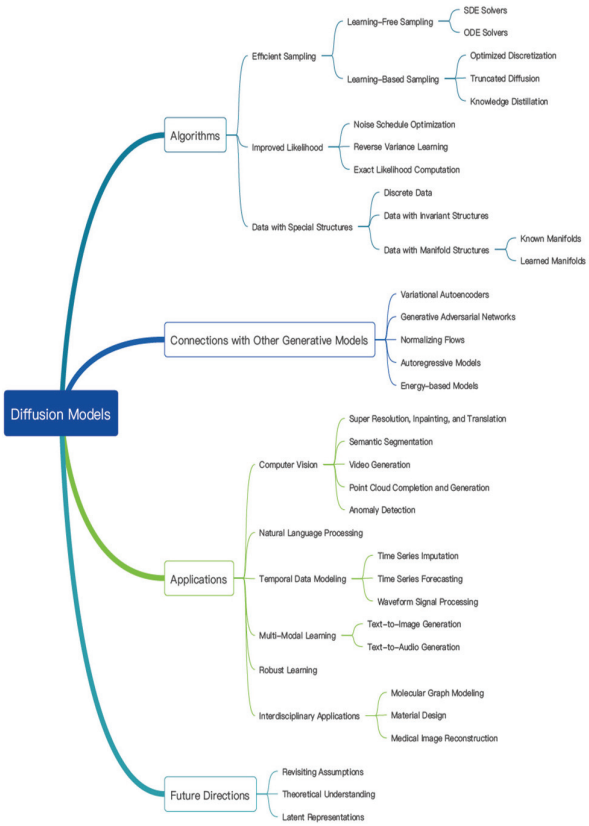

Figure 1 Landscape of deep generative models, highlighting the position of latent diffusion models (LDMs).

also making explicit the conceptual and practical links across these paradigms (e.g., score-based/SDE viewpoints, conditioning mechanisms, and the use of learned representations). The figure further organizes the diffusion-model literature by core algorithmic dimensions – sampling efficiency (learning-free solvers vs. learning-based acceleration/distillation), architectural improvements, and adaptations to structured or manifold-valued data – and by a broad spectrum of application areas spanning computer vision, language and multimodal generation, temporal data modeling, robustness-oriented learning, and interdisciplinary domains such as life sciences and medical imaging. Within this landscape, LDMs occupy a particularly important practical position: they preserve the strong quality and diversity properties of diffusion models while leveraging perceptually meaningful latent spaces to reduce computational cost, thereby enabling high-resolution generation that is feasible for large-scale and interactive deployments [2, 3, 18].

Diffusion models offer a complementary approach by learning to iteratively denoise noisy samples toward the data distribution, often yielding superior sample quality and diversity at the cost of many denoising steps. The technical foundations and variants of diffusion models are extensively summarized in [2–4], and [18].

Table 1 summarizes the dominant deep generative model families in terms of their training objectives, empirical strengths and weaknesses, and the deployment implications that matter in practice. VAEs and normalizing flows provide principled likelihood-based training and useful latent representations,

Table 1 Overview of deep generative model families

Model Family	Objective/Training Signal	Strengths	Limitations	Typical Web Applications
VAE	ELBO (reconstruction + KL)	Likelihood, latent structure	Blurry samples	Representation learning, compression
GAN	Adversarial loss	Sharp images, high realism	Mode collapse, unstable training	Image generation, style transfer
Autoregressive	Next-token/pixel likelihood	Exact likelihood, flexible	Slow sampling	Language, some images/audio
Normalizing flows	Exact likelihood via invertible maps	Exact likelihood	Architectural constraints	Density estimation
Diffusion	Denoising/score matching	High quality & diversity	Many sampling steps	Images, audio, etc.
LDM	Diffusion in learned latent space	Efficient, high resolution	Requires good autoencoder	Web-scale image/video generation

which makes them attractive for compression, representation learning, and density estimation, but they often face a quality gap (VAEs) or architectural rigidity (flows). GANs have historically delivered sharp, realistic samples and enabled many web-facing creative workflows, yet their adversarial objectives can lead to mode collapse and unstable training. Autoregressive models offer flexible, exact-likelihood formulations and have been especially influential for language, but their strictly sequential generation makes high-throughput, low-latency sampling challenging at scale. Diffusion models – and, more recently, LDMs – shift this trade-off frontier by prioritizing sample quality and diversity via denoising/score-matching objectives, at the cost of iterative sampling; LDMs mitigate this cost by performing diffusion in a learned latent space, making high-resolution generation substantially more feasible for large-scale services and interactive applications when paired with acceleration techniques such as improved samplers and distillation [1–3, 18].

2.2 Diffusion Models: Forward and Reverse Processes

Classical denoising diffusion probabilistic models (DDPMs) define a forward process that gradually adds Gaussian noise to data, forming a Markov chain from data to nearly isotropic Gaussian noise. A neural network learns a reverse process that denoises step by step. Score-based generative modeling frames a similar idea in terms of stochastic differential equations and score matching.

Key design dimensions – including noise schedules, parameterization (predicting noise, clean sample, or velocity), variance choices, and sampling schemes – have been systematized in recent technical surveys such as [2–4], and [18].

Table 2 summarizes the principal "knobs" that define the behavior of diffusion models and, by extension, LDMs-across both training and inference. In practice, these dimensions are tightly coupled: the noise schedule and variance treatment shape optimization stability and the effective difficulty of denoising at different timesteps, while the chosen parameterization (e.g., predicting ϵ, x, or v) influences loss scaling, gradient conditioning, and downstream sampling behavior. Guidance mechanisms, especially classifier-free guidance and multi-condition variants, provide a controllable trade-off between conditional alignment (e.g., prompt faithfulness) and sample diversity, but can also increase sensitivity to prompts and hyperparameters. Finally, the sampling strategy – ranging from ancestral DDPM sampling to deterministic DDIM, high-order solvers (e.g., DPM-Solver), and

Table 2 Design dimensions of diffusion and LDMs

Dimension	Options/Examples	Impact on Model Behavior	References
Noise schedule	Linear, cosine, sigmoid, custom	Trade-off between stability and quality	[2–4, 18]
Parameterization	ε-prediction, x-prediction, v-prediction	Affects loss scale and sampling	[2–4, 18]
Variance	Fixed, learned, hybrid	Controls stochasticity at each step	[2–4, 18]
Guidance	Classifier, classifier-free, multi-cond.	Alignment vs. diversity, prompt sensitivity	[2–4, 18]
Sampling strategy	DDPM, DDIM, DPM-solver, distillation	Sampling speed vs. quality	[2–4, 18]

distillation – largely determines the quality–latency–compute frontier: fewer steps improve throughput and responsiveness but can degrade fine details or increase artifacts if not carefully tuned. These system-level trade-offs have been extensively organized in recent diffusion surveys [2–4, 18] and motivate why implementation choices in LDM deployments are often driven as much by efficiency constraints as by purely generative quality considerations.

2.3 Latent Representations and Perceptual Compression

Training diffusion directly in pixel space for high-resolution images (e.g., 1024×1024 and beyond) is extremely expensive. A natural idea is to first learn a compressed latent representation via an autoencoder (often a VAE with perceptual and adversarial reconstruction losses) and then perform diffusion in that latent space, as proposed in [1].

The encoder E maps an image x to a latent representation $z = E(x)$ in a lower-dimensional space, while the decoder $\mathbb{D}$ reconstructs $\hat{x} = \mathbb{D}(z)$. Training usually combines pixel-wise losses with perceptual losses (e.g., feature-space distances) and, in some cases, adversarial losses to encourage sharpness and realistic textures, as described in [1]. The latent space is chosen to balance:

- Compression: Reduce spatial dimensions and channels for efficient diffusion.
- Perceptual fidelity: Preserve semantic and structural information relevant to downstream generation and editing [1, 7].

Figure 2 illustrates the core architectural decomposition that makes latent diffusion practical at high resolution. Instead of running the forward and reverse diffusion processes in pixel space, an encoder–decoder autoencoder

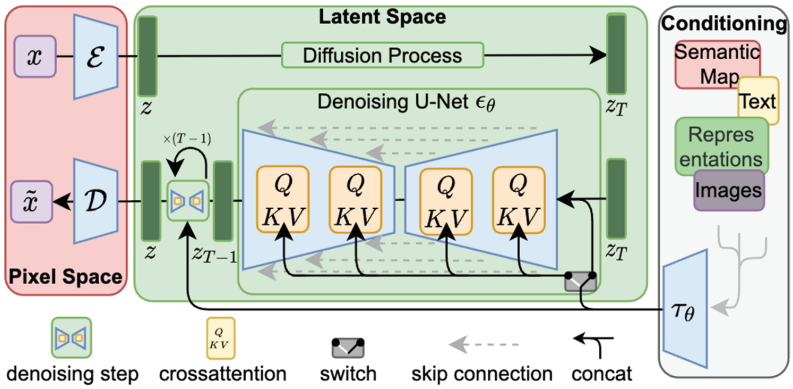

Figure 2 Conceptual architecture of a latent diffusion model (LDM). An autoencoder (E, D) compresses images from pixel apace into a perceptually meaningful latent space, where the diffusion process and denoising U-Net operate with cross-attentional conditioning.

(E, D) first maps an input image x into a lower-dimensional latent representation $z = E(x)$, and the denoising process is then executed on noisy latent variables z_t using a U-Net parameterized by ε_θ. This shift yields a large reduction in compute and memory footprint because the iterative denoising steps operate on compressed spatial grids, while the decoder D reconstructs the final image $\hat{x} = D(z_0)$ only once at the end. The figure also highlights the standard conditioning pathway used in many LDM systems: external signals (e.g., text embeddings, semantic maps, or reference images) are injected into the U-Net via cross-attention modules, enabling controllable generation and editing without changing the diffusion objective itself. Finally, the skip connections and concatenations shown in the U-Net block emphasize that, although the diffusion occurs in latent space, the model retains multi-scale feature aggregation – an important factor for preserving global structure while synthesizing fine-grained details during the denoising trajectory [1].

In video LDMs, this idea extends to spatio-temporal autoencoders that compress both spatial and temporal dimensions, as shown in [6] and the Stable Video Diffusion family described in [8]. In medical imaging and privacy-preserving setups, autoencoders can be tailored to domain-specific structures and anonymization requirements, as illustrated in [9] and [11].

2.4 From Pixel-space Diffusion to Latent Diffusion Models

The term "latent diffusion models" was formalized in [1], which showed that performing diffusion on compressed latents drastically reduces memory and

computes while retaining or improving visual quality. The LDM formulation combines:

1. A high-quality autoencoder providing perceptual compression and reconstruction [1].
2. A U-Net diffusion backbone operating in latent space [1, 7].
3. Cross-attention layers that inject conditioning signals (e.g., text embeddings, class labels, bounding boxes) [1, 7].

This design underpins Stable Diffusion and its derivatives, which have become standard building blocks in web-exposed generative services and developer APIs [1, 8, 19, 24].

3 Architectural Design of Latent Diffusion Models

3.1 Autoencoder Backbone

The autoencoder in an LDM defines the latent space in which diffusion operates, and its design strongly influences both reconstruction fidelity and computational cost. Many practical LDMs adopt spatial downsampling factors of 4 or 8 to strike a balance between compression and reconstruction quality [1, 7]. The latent dimensionality, i.e., the number of channels in the latent tensor, is another key degree of freedom: increasing the channel count typically improves visual fidelity but also raises the cost of diffusion. Training objectives usually combine pixel-wise reconstruction losses (L1/L2) with perceptual losses such as LPIPS and patch-based adversarial losses to encourage sharp, realistic reconstructions [1]. In addition, various regularization mechanisms are employed, including Kullback–Leibler divergence in VAE-style encoders, noise injection, and quantization in vector-quantized (VQ) variants, to stabilize training and shape the structure of the latent space.

In video LDMs, the autoencoder is extended to encode spatio-temporal volumes and decode them back to video, as discussed in [6] and [8]. In medical imaging, domain-specific autoencoders trained on modality-specific data (MRI, CT, X-ray) have been used as backbones for LDMs [9–14, 21].

Table 3 provides a representative cross-section of widely used LDM families and highlights how architectural choices co-vary with target resolution and application domain. A consistent pattern is that higher output resolutions are enabled primarily by (i) operating diffusion on aggressively downsampled latents (e.g., $64 \times 64 \times 4$ latents for 512×512 images) and (ii) scaling the denoising backbone – moving from standard attention-augmented U-Nets to

Table 3 Representative LDM architectures

Model/Family	Latent Resolution & Channels	Backbone (U-Net/DiT)	Target Resolution	Main Application Domain	Reference
Stable Diffusion v1	$64 \times 64 \times 4$ (for 512 $\times$ 512 images)	U-Net with attention	512×512	General text-to-image	[1, 19]
Stable Diffusion v2	...	U-Net	768×768	Higher-res text-to-image	[1, 19]
SDXL	Multi-scale latents	Larger U-Net + DiT blocks	1024×1024+	High-res, complex prompts	[7, 8, 19]
Latent video diffusion	Spatio-temporal latents	3D U-Net/temporal attention	Video frames	Video generation	[6, 8]
Med LDM (brain MRI)	3D latents	3D U-Net	3D volumes	Medical image synthesis	[9–11]

larger-capacity variants and, increasingly, hybridizations with transformer-style blocks (DiT-like components) to better model long-range dependencies for complex prompts. The table also emphasizes that "LDM" is not a single fixed architecture but a design template that generalizes across data modalities: for video, the same latent-diffusion principle is paired with spatio-temporal latents and 3D/temporal attention backbones, while in medical imaging the backbone and latent representation are adapted to volumetric structure and modality-specific priors. Overall, the entries illustrate the central trade-off surfaced in Section 3.1: increasing latent capacity and backbone scale tends to improve fidelity and controllability, but it directly increases the computational cost of iterative denoising – making the choice of latent resolution, channel count, and backbone family a first-order systems decision in real deployments [1, 7, 8, 19].

3.2 Latent Diffusion Backbone: U-Nets and Diffusion Transformers

Most LDMs adopt a U-Net-style architecture as the diffusion network. The original LDM employs a multi-scale U-Net with residual blocks, attention at lower resolutions, and cross-attention blocks for conditioning [1]. Subsequent models, such as SDXL, expand this backbone by increasing network depth and width, adding more attention layers to better handle complex prompts and high-resolution images [1, 7]. In parallel, diffusion transformers (DiT) have been introduced, in which transformer blocks operate on latent patches to provide improved scalability for large datasets and multimodal conditioning [7, 18]. For video LDMs, the diffusion backbone is further extended with temporal attention layers and spatio-temporal convolutions to capture temporal coherence across frames and maintain consistency over time [6, 8].

3.3 Conditioning Mechanisms

LDMs support flexible conditioning mechanisms that are central to many web applications. In text-to-image systems such as Stable Diffusion, text conditioning is implemented via cross-attention, where CLIP-like text embeddings are injected at multiple depths of the U-Net through cross-attention layers [1, 7], and these implementation patterns are reflected in production pipelines such as those described by the Diffusers library [19]. To control the trade-off between alignment and diversity, modern models commonly employ classifier-free guidance, which generates both conditioned and unconditioned predictions and interpolates between them; this strategy is widely used in

SDXL and related models and is surveyed in [2, 3], and [18]. Beyond pure text conditioning, many web-based editing tools rely on latent editing and inversion, in which invertible encoders and tailored noise scheduling map existing images into the latent space to enable operations such as background replacement, localized edits, and style transfer [1, 8, 19]. In domain-specific settings, such as medical imaging, LDMs (e.g., MedFusion and brain imaging LDMs) can additionally condition on clinical attributes, segmentation masks, or scanner parameters to better capture task-specific constraints.

3.4 Training Objectives and Noise Schedules

Training an LDM typically involves optimizing a denoising loss over a range of diffusion timesteps, and several design choices in this process strongly affect performance. A first choice is the noise parameterization: models may be trained to predict the injected noise (ε-prediction), the clean latent variable (x-prediction), or hybrid velocity-type quantities; comparative analyses of these parameterizations and their trade-offs are provided in [2, 3], and [4]. A second key component is the variance and noise schedule, where cosine or sigmoid schedules and learned variances are used to improve training stability and sample quality; these designs are broadly investigated in [2] and are summarized for practitioners in [18]. Finally, guidance strategies play a central role in aligning model outputs with desired conditions: classifier-based and classifier-free guidance, multi-condition guidance, and task-specific guidance mechanisms have all been explored, including approaches tailored for robust watermarking and safety constraints, as reviewed from a watermarking perspective in [5].

3.5 Sampling and Acceleration

Naïve diffusion sampling requires hundreds or even thousands of reverse steps, which is prohibitive for real-time or near-real-time web services. Consequently, practical LDM deployments rely heavily on sampling acceleration techniques. Deterministic samplers such as DDIM reduce the number of sampling steps while largely preserving image quality and are widely adopted in Stable Diffusion pipelines [2, 19]. Higher-order solvers, including DPM-solver-type integrators, further shrink the number of denoising steps (often to 10–30) with minimal degradation, and their design rationale is discussed in [4] and [18]. Distillation offers a complementary route, in which compact student models approximate the multi-step diffusion process in only a few steps; diffusion distillation methods are surveyed in [18] and are increasingly

integrated into production systems [8]. In addition, quantization and low-rank adaptation (LoRA) techniques reduce memory usage and enable efficient fine-tuning, which is particularly important for maintaining customized LDM instances in web applications [18, 19]. Together, these methods make efficient sampling a core enabling factor for deploying LDMs over web APIs at scale, where latency targets of only a few seconds per request are common.

4 Taxonomy of Latent Diffusion Models

We now categorize LDMs by data modality and application context, emphasizing representative models.

4.1 2D Image LDMs

The canonical LDM is the model of [1], which operates on 2D images through a latent autoencoder and a U-Net diffusion backbone with text conditioning. Stable Diffusion v1/v2 follow this design, differing in training data, resolution, and text encoders; their implementation details and pipelines are described in [1, 19], and the platform-oriented documentation in [8]. SDXL extends the architecture with larger capacity, multi-scale conditioning, and improved training recipes, enabling high-resolution synthesis suitable for production web services, as detailed in [7] and analyzed from a systems perspective in [8, 24]. These models serve as building blocks for many web applications and APIs.

4.2 Video and 4D LDMs

Latent video diffusion models extend LDMs to generate temporally coherent video by operating in a learned latent video space rather than directly on pixels. He et al. [6] introduce latent video diffusion models (LVDMs), which first learn a compact latent representation of video sequences and then apply diffusion in this space, achieving high-fidelity generation for long videos. Building on similar ideas, the Stable Video Diffusion family and related latent-video models are exposed through the Stability developer platform, providing video generation and editing capabilities via web APIs [8]. Recent "4D" extensions, which incorporate camera motion or 3D-aware representations, often combine LDMs with 3D priors and are discussed more broadly in diffusion-based vision surveys such as [3] and [4]. Compared with image-only LDMs, these video and 4D models face stricter memory and latency

constraints, which in turn motivate more aggressive compression schemes and sampling acceleration strategies, as emphasized in [6] and [8].

4.3 Domain-specific LDMs

Table 4 summarizes how the LDM paradigm has been specialized beyond general-purpose text-to-image generation to accommodate domain constraints, data structures, and task-specific objectives. In medical imaging, LDMs are frequently used for synthesis and reconstruction under strong regulatory and privacy requirements; consequently, model design often incorporates modality-specific autoencoders (e.g., 2D/3D latents for MRI/CT volumes), task-aware conditioning (segmentation masks, acquisition parameters, or anatomical priors), and governance-oriented constraints such as anonymization, auditability, and controlled deployment – features that differentiate these systems from open-ended creative generators [9–14, 21]. Watermarking-oriented LDM pipelines, by contrast, treat diffusion as a controllable transformation process where the central trade-off is robustness versus perceptual invisibility: the model and post-processing chain must preserve imperceptibility while maintaining reliable detection under common transformations (compression, resizing, mild edits), which motivates specialized objectives and evaluation protocols [5]. For text, diffusion-based generation typically requires bridging the discrete token space and continuous denoising dynamics (e.g., via continuous relaxations, embeddings, or latent-variable formulations), with the goal of retaining diffusion's parallelizable sampling advantages while achieving competitive linguistic quality [16]. In

Table 4 Domain-specific LDMs by modality

Domain/Modality	Typical Tasks	Example Models/Works	Notes	References
Medical imaging	Synthesis, reconstruction, anonymization	Brain MRI LDM, MedFusion, anonymization	Regulatory & privacy constraints	[9–14, 21]
Watermarking	Invisible watermarking, detection	LDM watermarking pipelines	Robustness vs imperceptibility	[5]
Text	Non-autoregressive text generation	Text diffusion models	Discrete to continuous mapping	[16]
Time series	Forecasting, imputation	Time-series diffusion with latents	Long-range dependencies	[17, 20]
Inverse problems	Deblurring, super-resolution, CT	LDM priors in inverse problems	Data-consistency constraints	[15]

time-series settings, LDMs are adapted to capture long-range dependencies and uncertainty in forecasting and imputation, often combining temporal encoders with latent diffusion to balance expressivity and computational efficiency on long horizons [17, 20]. Finally, for inverse problems such as deblurring, super-resolution, and CT reconstruction, LDMs are increasingly used as powerful learned priors integrated with data-consistency constraints; here, the dominant design question is how to couple the generative prior with the forward measurement model to achieve faithful reconstructions while remaining stable and efficient at inference time [15].

4.3.1 Medical imaging

LDMs have been actively explored in medical imaging. Pinaya et al. [9] introduce a brain-imaging generation pipeline using LDMs that produce realistic 3D MRI samples for data augmentation and analysis. Müller-Franzes et al. [10] show that diffusion probabilistic models can outperform GANs on a range of medical image synthesis tasks, including models that operate in latent spaces. Campos et al. [11] propose LDMs for privacy-preserving medical image anonymization, demonstrating that carefully designed latents can remove identifiable information while preserving clinical utility.

Several surveys synthesize diffusion-based methods for medical imaging. [12, 13], and [14] review diffusion models for medical image analysis, covering applications such as reconstruction, segmentation, synthesis, and domain adaptation. A curated list of diffusion models in medical imaging is maintained in [21]. These works highlight LDM variants for MRI, CT, X-ray, and ultrasound, including models that serve as priors in inverse problems and in domain-adaptation pipelines [12–15, 21].

4.3.2 Watermarking and provenance

Hur et al. [5] review latent diffusion models for image watermarking, surveying recent trends in embedding and detecting watermarks in LDM outputs. LDMs can incorporate watermarking directly in latent space or in image space, balancing robustness and imperceptibility; these mechanisms are particularly relevant to web-exposed generative services that need to signal AI-generated content and support provenance [5].

4.3.3 Text and discrete data

Li et al. [16] survey diffusion models for non-autoregressive text generation, including latent-space diffusion approaches where discrete tokens are

mapped to continuous representations before diffusion. While many of these models are not classical LDMs in the sense of [1], similar latent-space diffusion ideas appear in compressing and modeling discrete sequences and are also discussed in broader diffusion surveys such as [18].

4.3.4 Time series and spatio-temporal data

Yang et al. [17] present a survey on diffusion models for time series and spatio-temporal data, covering applications such as forecasting, imputation, and anomaly detection. Again, while not all models are strict LDMs, several approaches use learned latent spaces to handle long time horizons and complex dependencies. A curated list documenting diffusion models for time series is maintained in [20], providing code and dataset links that are useful for web-scale experimentation.

4.3.5 Inverse problems

Daras et al. [15] survey diffusion models for inverse problems (e.g., deblurring, super-resolution, tomography), including LDM-style priors that operate in compressed feature spaces. LDMs are particularly appealing where domain-specific autoencoders already exist (e.g., MRI, CT) and can be reused for latent diffusion [10, 12, 15].

4.4 Ecosystem and Tooling

Open-source libraries and curated resources have played a central role in accelerating the adoption of LDMs. The original latent-diffusion and Stable Diffusion repositories described in [1] and subsequently maintained via the diffusion pipelines of [19] provide reference implementations, pretrained checkpoints, and configuration templates that many later works build upon. On top of these foundations, Hugging Face Diffusers offers unified, production-oriented pipelines for Stable Diffusion and related models, with a strong emphasis on deployment, optimization, and interoperability across hardware backends [19]. In parallel, curated lists such as those compiled by [20] for time-series applications and by [21] for medical imaging catalog tasks, datasets, and codebases, thereby supportreproducible research and engineering practice. Collectively, these open resources shape how LDMs are integrated into web services and platforms – from rapid prototyping in notebooks to large-scale serving in production environments – as exemplified by [19, 23], and [8, 24].

5 Evaluation Methodologies

Evaluating LDMs involves both model-centric and system-centric metrics. Surveys on diffusion models, such as [2–4, 12], and [18], provide overviews of common evaluation practices, which we adapt here to LDM-specific and web-centric contexts.

5.1 Image and Video Quality Metrics

Common quality and diversity metrics for LDMs largely follow those used in the broader generative modeling literature. The Fréchet inception distance (FID) and the Kernel inception distance (KID) quantify distributional similarity between generated and real images and have become the dominant benchmarks in diffusion studies summarized in [2, 3], and [18]. The inception score (IS) also estimates the quality and diversity of generated samples, although it has gradually been supplanted by FID and KID in recent diffusion research [2]. For text-to-image models, CLIP-based scores such as CLIPScore evaluate text–image alignment, which is critical for prompt-driven LDMs; these metrics are discussed in [3] and [18] and are commonly used in Stable Diffusion evaluations [19]. In the video domain, temporal metrics including Fréchet video distance (FVD) and temporal LPIPS are employed to assess temporal consistency and perceptual quality across frames [3, 6, 18].

Table 5 highlights that no single metric is sufficient to characterize LDM performance, because different metrics capture different failure modes. In practice, FID and KID are most often used as primary "offline" indicators of realism and diversity, but both depend on the chosen feature extractor and can

Table 5 Evaluation metrics for LDMs

Metric	Type (Quality/) (Diversity/) Alignment/etc.)	Description	Strengths	Limitations	References
FID	Quality & diversity	Distance between feature distributions	Widely used, interpretable	Sensitive to feature extractor	[2, 3, 18, 19]
KID	Quality & diversity	MMD-based kernel distance	Unbiased estimate	Less standardized than FID	[2, 3, 18]
Inception score	Quality & diversity	Classifier-based score	Simple to compute	Not robust, dataset dependent	[2, 3, 18]
CLIPScore	Text–image alignment	CLIP similarity between text and image	Measures semantic alignment	Depends on CLIP biases	[3, 18, 19]
FVD	Temporal quality (video)	Distance between video feature distributions	Evaluates temporal coherence	Requires video datasets	[3, 6, 18]

be sensitive to evaluation protocol details (e.g., the number of generated samples, preprocessing, and dataset mismatch), which complicates direct cross-paper comparisons [2, 3, 18]. Inception score remains historically influential and easy to compute, yet it is particularly dataset- and classifier-dependent and can be gamed by producing confident-but-narrow samples; consequently, recent diffusion work tends to treat it as secondary to FID/KID [2, 3]. For prompt-driven LDMs, CLIPScore (and related CLIP-based measures) is valuable because it targets text–image alignment, but it also inherits biases and blind spots of the underlying CLIP model and does not reliably capture photographic quality, fine detail, or safety-related attributes; thus, it is best interpreted jointly with quality/diversity metrics and qualitative inspection [3, 18, 19]. In the video setting, FVD and temporal perceptual metrics extend the same logic to spatio-temporal coherence, but they are computationally heavier, require carefully curated video datasets, and can under-penalize certain artifacts (e.g., repeated textures or subtle temporal flicker) unless paired with complementary temporal-consistency checks [3, 6, 18]. Overall, Table 5 motivates a multi-metric evaluation stack – typically combining distributional similarity (FID/KID), conditional alignment (CLIP-based), and temporal coherence (FVD/temporal LPIPS) – with transparent reporting of protocols (sample counts, prompts, dataset splits) to ensure that results are reproducible and meaningful for real-world deployment decisions [2, 3, 18].

5.2 Task-specific and Downstream Metrics

Domain-specific LDMs are often evaluated indirectly through their impact on downstream tasks rather than solely by generic perceptual scores. In medical imaging, for example, LDM-generated images are assessed by how well they support segmentation, detection, or reconstruction pipelines; typical metrics include Dice coefficients for segmentation, detection precision and recall, and reconstruction error, as used in [9–11], and the surveys [12–14]. For text and time-series applications, evaluations likewise rely on task-specific criteria such as BLEU and ROUGE scores and perplexity for text, as well as forecasting error (MAE/RMSE) and anomaly-detection AUC for temporal data, following the discussions in [16, 17], and [20]. In web-deployed systems, these offline metrics are typically complemented by online measurements obtained from A/B testing, click-through rates, and other engagement indicators, which provide additional insight into user satisfaction and business impact, as emphasized in platform documentation such as [8, 24] and in prior web-systems survey discussions [26, 27].

Table 6 Domain-specific/downstream evaluation

Domain	Downstream Task	Metrics (Examples)	Example Works	Notes
Medical imaging	Segmentation from synthetic data	Dice, IoU, HD95	[9–14]	Measure if synthetic data improves training
Medical imaging	Reconstruction	PSNR, SSIM, NMSE	[10, 12–15]	LDM as prior for inverse problems
Text	Generation quality	BLEU, ROUGE, human eval	[16]	Fluency vs diversity trade-offs
Time series	Forecasting	MAE, RMSE, MAPE	[17, 20]	Multi-step forecast performance
Inverse problems	Super-resolution/ deblurring	PSNR, SSIM, perceptual metrics	[15]	Data-consistent reconstructions

Table 6 emphasizes that, for domain-specific LDMs, evaluation is often most meaningful when framed in terms of utility – i.e., whether generated or reconstructed outputs measurably improve an end task – rather than solely in terms of generic perceptual similarity. In medical imaging, this typically means assessing whether LDM-generated data improves segmentation or detection models (e.g., via Dice/IoU/HD95 for segmentation, precision–recall metrics for detection), or whether LDM-based priors improve reconstruction fidelity under limited or noisy measurements (e.g., PSNR/SSIM/NMSE), since these metrics align directly with clinical or operational objectives and can reveal failures that perceptual scores miss (e.g., anatomically implausible structures that still look "realistic") [9–15]. For text and time-series diffusion models, downstream metrics likewise target task performance: BLEU/ROUGE and human preference judgments probe semantic adequacy and fluency in text generation, while MAE/RMSE/MAPE and anomaly-detection AUC capture predictive accuracy and decision relevance in temporal settings [16, 17, 20]. Importantly, Table 6 also underscores a deployment reality: in web-facing products, offline task metrics are rarely sufficient on their own, and are typically paired with online experimentation (A/B tests, CTR, retention, satisfaction proxies) to quantify user impact and business value under real traffic distributions, where latency, reliability, and safety constraints interact with model quality [8, 24, 26, 27]. Consequently, rigorous evaluation of LDM systems is best viewed as a two-layer process – offline task-validity checks for scientific comparability, followed by online measurement to validate usefulness and robustness in the operational environment.

5.3 Safety, Robustness, and Watermarking

Safety and robustness are particularly important for LDMs exposed via the web [5, 9–14, 18, 28]. In safety-critical and clinical scenarios, evaluation increasingly focuses on several complementary aspects. First, content-moderation performance is assessed by measuring a model's ability to detect NSFW or otherwise harmful content and by quantifying how easily prompt-based "jailbreaking" can occur when models are repurposed or misused [10, 12–14]. Second, robustness to adversarial edits is evaluated by testing whether prompt manipulations can bypass safety filters, especially in hosted services where users can iteratively refine prompts [10, 11, 18]. Third, for watermarking schemes embedded in LDM outputs, robustness is measured in terms of detection accuracy under common transformations such as resizing, compression, cropping, and adversarial removal, as surveyed in [5]. These considerations are particularly important for web platforms that serve large, diverse user bases and must comply with regulations on medical and personal data [11–14, 18, 25].

6 LDMs in Web Engineering: Systems and Applications

A key requirement for articles in web engineering venues is that the web-engineering relevance be explicit [25]. LDMs are now deeply integrated into web ecosystems as services, APIs, and end-user applications [8, 19, 22–24].

6.1 Web-exposed LDM Services

Several public platforms now expose LDMs as web services. Hosted web user interfaces built with frameworks such as Gradio allow users to generate images directly from their browser on managed infrastructure, often running on Hugging Face Spaces or similar hosting platforms [19, 23]. Beyond interactive UIs, third-party Stable Diffusion APIs – for example, services such as stablediffusionapi.com – offer REST endpoints for image generation, upscaling, and editing, targeting mobile and web developers who wish to integrate LDM capabilities programmatically into their applications [22]. At a larger scale, the Stability AI developer platform provides a unified API for multiple LDM-based models (including Stable Diffusion v3.x, SDXL, and Stable Video Diffusion), together with documentation on authentication, pricing, and integration that facilitates deployment in commercial web applications [8, 24]. Collectively, these platforms exemplify typical web-engineering concerns – API design, authentication and authorization, rate limiting, request

Table 7 LDM-based platforms and APIs

Platform/ Service	Access Type (UI/API)	Supported Models (Examples)	Key Features	Reference
Stability API	REST API	SDXL, SD 3.x, Stable Video	Auth, pricing, usage limits	[8, 24]
stablediffusionapi.com	REST API	Stable Diffusion variants	Simple API, multiple endpoints	[22]
Hugging Face Spaces	Web UI + API	WebUI apps for Stable Diffusion	Community-hosted UIs, Gradio	[23]
Diffusers library	Python library + CLIs	Many diffusion/LDM pipelines	Unified API, optimization	[19]

tracing, logging, and billing – which mirror themes discussed in prior system-oriented surveys in the web engineering literature [26, 27].

Table 7 consolidates representative access patterns through which LDM capabilities are operationalized as web-facing products, ranging from end-user interfaces to developer-oriented APIs and software libraries. A common distinction is between hosted interactive experiences (e.g., browser-based generation UIs deployed on community or managed hosting) and programmable service layers that expose generation, editing, and upscaling as REST endpoints for integration into mobile and web applications [Diffusers Team, 19; Hugging Face Spaces, 23; stablediffusionapi.com, 22]. The Stability API illustrates a third pattern – a commercial platform API that standardizes authentication, pricing, quota enforcement, and versioned model access across multiple LDM families (e.g., SDXL, SD 3.x, and video variants), effectively treating model inference as a metered cloud service [8, 24]. From a systems standpoint, the table underscores that "model quality" is only one component of deployment readiness: production offerings must also provide predictable interfaces, backward-compatible model/version management, observability (request tracing and logging), reliability controls (timeouts, retries, queueing), and governance mechanisms (safety filters and watermarking where applicable). These platform concerns align closely with recurring themes in system-level surveys of web search and question answering, where the engineering challenge lies in wrapping sophisticated models inside robust, measurable, and economically sustainable service abstractions rather than in the model alone [26, 27].

6.2 Application Patterns in Web Systems

Web applications employ LDMs in several recurring patterns that reflect different modes of interaction and integration. A first class of use cases

comprises interactive creative tools, including text-to-image user interfaces for marketing creatives, game art, and UI mock-ups, as well as image-to-image editors that support style transfer, background replacement, and product photography. A second class centers on programmatic content generation, where content-management or e-commerce backends automatically generate or modify visual assets for large catalogs of items, and ad-tech pipelines produce and test multiple creatives via LDMs while optimizing them based on user engagement. A third pattern involves domain-specific portals, such as web platforms that provide synthetic medical images for research and education – sometimes incorporating LDM-based anonymization [9, 11, 21] – and scientific visualization services that rely on LDMs trained on specialized datasets. Finally, developer-facing platforms expose LDM capabilities through API gateways and software development kits, allowing web developers to integrate generative features with minimal machine-learning expertise by wrapping Diffusers-based pipelines or commercial APIs [8, 19, 22, 24].

6.3 System Architecture for LDM-based Web Services

Typical architecture patterns for web-deployed LDMs largely follow modern microservice-based designs. A browser-based frontend or single-page application communicates with a backend gateway that routes requests to inference services hosting LDM models on GPU nodes, while auxiliary services handle logging, metrics collection, watermark insertion and verification, safety filtering, and caching; comparable multi-tier architectures have been described in prior surveys of web search and question-answering systems [26, 27]. Model-serving strategies depend on workload characteristics: offline or dataset-generation workloads can exploit batching for efficiency, whereas interactive user interfaces typically require per-request inference. In addition, models are often sharded across GPUs or container instances to expose different variants (e.g., SDXL, SD 3.x, Stable Video Diffusion) under a unified service layer [7, 8]. To handle bursty demand in public web services, horizontal autoscaling based on GPU utilization and request-queue length is essential [8, 24]. Latency management is also a central concern: low-step samplers and distilled student models are used to keep synchronous image-generation flows within target response times (often around 1–3 seconds per image), whereas longer-running tasks such as video generation are commonly shifted to asynchronous workflows with status polling or notification callbacks [6, 8].

Table 8 System-level design patterns for LDM web services

Pattern	Description	Advantages	Drawbacks/ Trade-offs	References
Microservice architecture	Separate gateway, inference, logging	Scalability, modularity	Operational complexity	[8, 24, 26, 27]
Autoscaling	Scale GPU nodes by load	Cost efficiency, burst handling	Scaling delay, cold starts	[8, 24, 18]
Asynchronous jobs	Queue-based long-running generation	Handles long tasks (video)	More complex client logic	[8, 24, 26, 27]
Batching & caching	Batch similar requests, cache results	Higher throughput, lower cost	Latency for individual requests	[18, 24]

Table 8 distills recurring system-level patterns that determine whether an LDM-backed service is merely "deployable" or truly production-grade under real web traffic. The microservice decomposition (gateway, inference, and auxiliary observability/safety components) improves modularity and independent scaling, but it introduces operational overhead – interface versioning, service discovery, distributed tracing, and incident response become first-order concerns once requests traverse multiple hops [8, 24, 26, 27]. Autoscaling is essential for handling bursty demand while controlling GPU cost; however, it must be engineered around practical constraints such as cold-start latency (container/model load times), GPU fragmentation, and queue instability during sudden traffic spikes [8, 24, 18]. The asynchronous job pattern is a natural fit for long-running generation (notably video), where synchronous HTTP timeouts and interactive UX constraints make queue-based execution with polling or callbacks the dominant design; the trade-off is increased client and product complexity, including progress reporting, idempotency, cancellation, and retry semantics [8, 24, 26, 27]. Finally, batching and caching directly influence unit economics: batching amortizes GPU overhead across requests and can substantially increase throughput for offline or high-volume workloads, while caching (at the prompt/seed/output level, or via intermediate features) reduces redundant compute; both techniques, however, must be balanced against latency SLOs, personalization needs, and the risk of serving stale or policy-inconsistent outputs when safety filters or model versions change [18, 24]. Taken together, the patterns in Table 8 formalize the central deployment trade-off for web-facing LDMs: maximizing quality and controllability while meeting latency, reliability, and cost

constraints through careful orchestration of scaling, execution mode (sync vs async), and inference optimization.

6.4 Resource Efficiency and Cost

Because LDMs remain computationally intensive, web engineering for LDM-based services must explicitly account for GPU resource management, caching, and configuration control. GPU sharing and scheduling become nontrivial when mixed workloads – such as small versus large images or still images versus video – compete for the same accelerators. Caching can mitigate some of this cost by storing frequently requested prompts or intermediate latent representations when privacy constraints permit, leveraging the substantial repetition observed in public web UIs [8, 19, 23]. At the same time, exposed inference parameters (e.g., number of diffusion steps, guidance scale, and output resolution) must be bounded and validated to prevent unbounded growth in latency and infrastructure cost. Diffusion-efficiency techniques originally studied in medical imaging – such as lightweight architectures, pruning, and knowledge distillation [12–14] – are equally relevant for web deployment. Similar considerations arise in time-series and inverse-problem settings, where diffusion models are required to operate under tight compute budgets while still meeting task-specific accuracy requirements [15, 17, 20].

7 Safety, Ethics, and Governance in Web-exposed LDMs

LDMs deployed on the web raise legal, ethical, and societal challenges [5, 11–14, 18, 25, 28, 29].

7.1 Copyright and Training Data

Stable Diffusion and similar models have faced legal challenges over the use of copyrighted training data. A recent UK case between Getty Images and Stability AI, reported in [29], led to findings of trademark infringement for images including Getty watermarks, while broader questions about copyright in model training remained unresolved. This uncertainty affects both model developers and web platforms consuming these APIs. Web-scale deployment must therefore consider licensing of training data, indemnification clauses, and clear documentation of content usage, as emphasized in [8, 24] and broader governance discussions in [18] and [25].

7.2 Deepfakes and Harmful Content

Open-source LDMs such as Stable Diffusion and newer models like Flux can be fine-tuned into deepfake generators with minimal technical expertise. The Oxford Internet Institute [28] reports tens of thousands of such generators hosted on platforms including Civitai and Hugging Face, collectively downloaded millions of times and predominantly used to create non-consensual explicit images, posing serious risks to privacy, harassment, and reputational harm. In response, web-scale LDM services must incorporate multiple layers of protection, including prompt filtering and safety classifiers, NSFW and face-matching detection systems with strong privacy safeguards, and clearly articulated community guidelines accompanied by reporting mechanisms and enforcement processes [8, 19, 23, 24, 28].

7.3 Watermarking, Provenance, and Detection

As LDM-generated content proliferates, provenance mechanisms become increasingly critical. Hur et al. [5] survey watermarking strategies tailored to latent diffusion models, including schemes that embed signals in either the latent or image domain. Building on these ideas, platform-level approaches typically combine several measures: embedded watermarks or invisible markers that explicitly denote AI-generated content [5]; metadata and cryptographic signatures attached at generation time via platform APIs, which support downstream verification and policy enforcement [8, 24]; and detection models trained to distinguish AI-generated samples from real data, even though adversarial training and model adaptation can erode the effectiveness of such detectors over time [5, 18, 28]. From a web-engineering perspective, watermarking and provenance must be integrated at the infrastructure level-during rendering, upload, and distribution-rather than being left solely to individual applications, an architectural requirement emphasized in both watermarking surveys and discussions within the Journal of Web Engineering [5, 25].

Table 9 summarizes the practical mechanism stack that platforms typically assemble to manage safety and provenance for LDM-generated content, and it highlights why these controls must be treated as end-to-end system properties rather than isolated model add-ons. Watermarking approaches differ primarily by where the signal is injected: latent-space embedding can be more robust to common post-processing and may be harder to remove without degrading content, but it can interact with the latent distribution and potentially affect generation fidelity; image-space watermarking is simpler

Table 9 Safety, watermarking, and deepfake mitigation techniques

Technique	Integration Point (Latent/ Image/Post)	Strengths	Limitations/Risks	References
Latent watermark embedding	Latent space during generation	Harder to remove, robust	May affect latent distribution	[5]
Image-space watermarking	Post-processing on generated images	Simple, model-agnostic	Easier to crop or modify	[5]
Metadata/ signatures	At API/platform layer	Clear provenance, audit trails	Can be stripped by re-encoding	[8, 18, 24]
Safety classifiers	Post-generation filtering	Blocks harmful/NSFW content	Adversarial prompts, false negatives	[11, 18, 28]
Deepfake detection	Model-based deepfake detectors	Detects synthetic faces/content	Arms race with generator models	[28, 29]

and model-agnostic, yet it is more vulnerable to removal via cropping, resizing, or re-encoding transformations [5]. Metadata and cryptographic signatures introduced at the API/platform layer provide a complementary provenance channel – enabling audit trails, accountability, and policy enforcement – but they rely on preservation of metadata across downstream pipelines, and thus can be weakened when content is redistributed through channels that strip or rewrite headers and EXIF-like fields [8, 18, 24]. Safety controls similarly require layered defenses: post-generation safety classifiers can reduce harmful or policy-violating outputs, but they remain susceptible to adversarial prompting and distribution shift; as a result, many deployments combine classifier-based filtering with rate limits, abuse monitoring, and human review workflows for high-risk cases [11, 18, 28]. Finally, deepfake detection illustrates an inherent arms race: detector performance can degrade as generators improve or adapt, implying that detection should be coupled with provenance signals and platform governance rather than treated as a standalone solution [28]. Taken together, the techniques in Table 9 motivate a defense-in-depth architecture in which watermarking, metadata, filtering, and detection are integrated into generation, storage, and distribution workflows to maintain verifiable provenance and enforce safety policies at web scale [5, 8, 24, 25].

7.4 Fairness, Bias, and Representation

LDMs trained on large web-scraped datasets inevitably inherit biases related to gender, race, culture, and social roles. Although this survey does not aim to provide an exhaustive review of fairness-mitigation techniques, several directions are particularly relevant. First, careful curation of training data and prompt templates is repeatedly emphasized in diffusion-model and broader AI survey literature as a primary line of defense against amplifying harmful stereotypes [2, 18]. Second, post-hoc debiasing strategies – such as targeted fine-tuning or modified guidance procedures – have been proposed in the research literature and, in some cases, reflected in platform guidelines as practical ways to mitigate biased behavior in deployed models [8, 24]. Finally, transparency toward end-users about model limitations and bias risks remains crucial, aligning with broader concerns around fairness and accountability in web search and recommendation systems discussed in the web engineering literature [26, 27].

8 Open Research Directions

We highlight several open directions at the intersection of LDM research and web-engineering practice, synthesizing perspectives from recent diffusion surveys, domain-specific LDM studies, and discussions on provenance and governance in web platforms [2–5, 9–18, 20, 25, 28, 29]. First, advances in controllable and interpretable LDMs – e.g., structured conditioning interfaces (scene graphs, UI layouts, and symbolic constraints) and more disentangled or interpretable latent subspaces – would enable developers to build generative services that behave predictably under diverse prompts and production traffic. Second, the rise of multimodal web ecosystems, where LDMs are composed with large language models, audio diffusion models, and 3D/4D generators for applications such as generative games and interactive storytelling, introduces new requirements for cross-modal consistency, interface design, and end-to-end safety management [3, 18]. Third, resource-efficient deployment remains central: progress in distillation, low-rank adaptation, quantization, dynamic routing, and multi-scale acceleration can substantially reduce latency and cost while expanding feasibility to edge or on-device settings [4, 15, 12–14, 18]. Fourth, standardized evaluation protocols tailored to web settings are needed; beyond offline perceptual scores, benchmarks that incorporate latency-adjusted quality, reliability, user-experience outcomes, fairness, and safety would support more principled engineering choices [2, 3,

Table 10 Open research directions for LDMs in web engineering

Direction	Short Description	Key References	Implications for Web Engineering
Controllable & interpretable LDMs	Better control signals and interpretable latents	[2–4, 18]	More predictable web services, safer UX
Multimodal web ecosystems	LDMs + LLMs + audio/3D models	[3, 18]	Rich interactive applications, new UI patterns
Efficiency & deployment	Distillation, quantization, LoRA, edge deployment	[4, 12–15, 18]	Lower cost, wider hardware support
Standardized evaluation	Benchmarks including safety & latency	[2, 3, 12–14, 18]	Comparable systems, better engineering choices
Legal & regulatory frameworks	Copyright, deepfakes, consent, liability	[25, 28, 29]	Compliance requirements for web platforms
Domain-specific LDM-as-prior	Integrating LDMs into medical/scientific pipelines	[9–12, 14, 15, 21]	Specialized portals, trusted domain services

12–14, 18]. Fifth, evolving legal and regulatory frameworks will increasingly shape deployment practices on the web, especially around data use, provenance and watermarking, consent, and liability [25, 28, 29]. Finally, in medical imaging and scientific computing, LDMs are emerging as powerful priors or conditional generators embedded in analysis pipelines and may be delivered through specialized web portals and APIs that support domain constraints, auditability, and privacy-preserving operation [9–15, 21].

Table 10 summarizes these directions and makes explicit their implications for web-engineering design. Controllability and interpretability translate into concrete product and API primitives – typed control signals, constraint-aware generation endpoints, and debuggable intermediate representations – that reduce unexpected behavior and improve operational reliability. Multimodal ecosystems, in turn, call for compositional service orchestration, shared safety policies across modalities, and interface patterns that keep users "in the loop" while maintaining cross-modal coherence. Efficiency-oriented research directly affects platform unit economics and SLO compliance by enabling faster inference, broader hardware support, and graceful degradation strategies under load. Standardized evaluation frameworks would allow teams to compare systems under consistent protocols that

reflect real deployments, integrating latency, robustness, fairness, and safety into decision-making rather than treating them as separate afterthoughts. Legal and regulatory evolution motivates compliance-by-design architectures – provenance capture, watermark/signature enforcement, auditable logs, and policy-aware content pipelines. Finally, domain-specific LDM-as-prior settings emphasize trusted delivery: curated portals and APIs that expose domain-appropriate controls, privacy guarantees, and validation hooks so that generative capabilities can be integrated into professional workflows with clear governance boundaries.

9 Conclusion

Latent diffusion models (LDMs) have become a central paradigm for high-fidelity generative modeling, particularly in settings where resolution, latency, and scalability constraints matter. By separating (i) perceptual compression via an autoencoder from (ii) iterative denoising in a learned latent space, LDMs substantially reduce the computational burden of diffusion while preserving strong sample quality – an engineering trade-off that makes large-scale image and video generation more feasible on contemporary GPU infrastructure [1, 7]. This practicality is reflected in widely used software stacks and platform deployments that expose LDM capabilities as reusable components in production content pipelines [8, 19, 24].

This survey consolidated the LDM landscape from first principles to deployment practice. On the algorithmic side, we organized key design dimensions – noise schedules, parameterizations, variance handling, guidance mechanisms, and sampling solvers – together with the architectural choices that dominate modern LDM families, including autoencoder backbones, cross-attention conditioning, and scaling strategies for higher resolution and more complex prompts [2–4, 18]. On the systems side, we framed these modeling choices in terms of the quality–latency–cost frontier that governs real services: compression ratios and latent capacity determine denoising cost; sampler and distillation choices determine responsiveness; and conditioning interfaces determine controllability and product reliability.

A consistent conclusion across the literature is that evaluation must be multi-layered. Offline perceptual and distributional metrics (e.g., FID/KID and alignment measures such as CLIP-based scores) remain useful for research comparability, but they are incomplete proxies for user value, robustness, and safety under real-world usage [2, 3, 18]. In domain settings,

the most informative signal often comes from downstream utility – e.g., whether generated data improves segmentation, detection, or reconstruction performance in medical imaging – because plausible-looking outputs can still be invalid for the task or violate domain constraints [9–15]. For web-facing systems, offline metrics are therefore best interpreted alongside online measurements (A/B tests, engagement, retention, and SLO-oriented reliability metrics) to ensure that gains persist under shifting traffic distributions and evolving user behaviors [8, 24, 26, 27].

From an engineering standpoint, LDMs increasingly function as web-exposed capabilities rather than standalone research artifacts. The survey highlighted recurring service patterns – API gateways routing to GPU-backed inference workers, auxiliary services for safety filtering and watermarking, observability and tracing, caching/batching for throughput, and asynchronous job orchestration for long-running tasks such as video generation – that collectively determine whether an LDM can be operated reliably and economically at scale [6, 8, 24]. Platform examples and tooling ecosystems illustrate that interface stability (versioning and backward compatibility), workload-aware serving (batch vs per-request inference; sync vs async flows), and operational governance (quotas, abuse monitoring, and audit logs) are as critical as model architecture for sustaining production deployments [8, 19, 22–24].

We also emphasized that responsible deployment requires treating safety, provenance, and governance as first-class system requirements. Bias inherited from web-scale datasets motivates careful data and prompt-template curation, as well as practical mitigations during fine-tuning and inference-time guidance, supported by transparent user communication about limitations and risks [2, 18]. Provenance mechanisms – watermarking, metadata/signatures, and detector-based monitoring – are most effective when integrated end-to-end across generation, upload, and distribution pipelines, since post hoc controls can be weakened by re-encoding and adversarial adaptation [5, 8, 18, 24, 28]. Finally, evolving legal and regulatory expectations around data use, consent, and liability reinforce the need for compliance-by-design architectures that support auditing, policy enforcement, and traceability [25, 28, 29].

Looking forward, the most impactful progress is likely to come from tighter co-design of models, evaluation, and infrastructure. On the modeling side, improved controllability and interpretability should be expressed as stable, typed interfaces (structured constraints, verifiable controls, and predictable failure modes) that can be monitored and tested in production,

rather than relying on ad hoc prompting [3, 18]. On the systems side, standardized benchmarks and reporting practices that jointly capture quality, latency, reliability, and safety would make web-deployed LDM systems more comparable and would accelerate the translation of algorithmic advances into trustworthy user experiences [2, 18]. In parallel, continued work on efficiency (distillation, quantization, low-rank adaptation, and scalable serving) and domain-specific integration (e.g., medical and scientific portals with explicit governance constraints) will broaden the range of platforms and contexts where LDMs can be deployed responsibly [4, 9–12, 15, 21].

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT)(IITP-2025-RS-2024-00436765).

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution image synthesis with latent diffusion models," Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695, 2022.

[2] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," ACM Computing Surveys, vol. 56, no. 4, article 105, 2024.

[3] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 45, no. 9, pp. 10850–10869, 2023.

[4] Z. Chang, G. A. Koulieris, and H. P. H. Shum, "On the design fundamentals of diffusion models: A survey," Pattern Recognition, vol. 169, article 111934, 2026.

[5] H. Hur, M. Kang, S. Seo, and J.-U. Hou, "Latent diffusion models for image watermarking: A review of recent trends and future directions," Electronics, vol. 14, no. 1, article 25, 2025.

[6] Y. He et al., "Latent video diffusion models for high-fidelity long video generation," arXiv preprint arXiv:2211.13221, 2022.

[7] D. Podell et al., "SDXL: Improving latent diffusion models for high-resolution image synthesis," Proc. Int. Conf. Learning Representations (ICLR), 2024; also arXiv preprint arXiv:2307.01952, 2023.

[8] Stability AI, "Stable Diffusion and Stable Video Diffusion developer platform," technical documentation and API reference, 2023–2025 (online, accessed 2025).

[9] W. H. L. Pinaya, M. S. Graham, E. Kerfoot, et al., "Brain imaging generation with latent diffusion models," in Deep Generative Models, DGM4MICCAI 2022, Lecture Notes in Computer Science, vol. 13609, pp. 117–126, Springer, 2022.

[10] G. Müller-Franzes, J. M. Niehues, F. Khader, et al., "Diffusion probabilistic models beat GANs on medical images," arXiv preprint arXiv:2212.07501, 2022; and "A multimodal comparison of latent denoising diffusion probabilistic models and generative adversarial networks for medical image synthesis," Scientific Reports, vol. 13, article 12098, 2023.

[11] F. Campos, M. A. T. Figueiredo, B. L. Póvoa, et al., "Latent diffusion models for privacy-preserving medical image anonymization," in Proc. 3rd Workshop on eXplainable AI in Healthcare (XAI-Healthcare), CEUR Workshop Proceedings, vol. 3831, 2024.

[12] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof, "Diffusion models for medical image analysis: A comprehensive survey," arXiv preprint arXiv:2211.07804, 2022; and "Diffusion models in medical imaging: A comprehensive survey," Medical Image Analysis, vol. 88, article 102846, 2023.

[13] Y. Shi, A. Abulizi, H. Wang, et al., "Diffusion models for medical image computing: A survey," Tsinghua Science and Technology, vol. 30, no. 1, pp. 357–383, 2025.

[14] Q. Liu, Y. Guan, W. Wu, H. Shan, and D. Liang, "Diffusion models in medical imaging: A comprehensive survey," CT Theory and Applications, vol. 34, no. 3, pp. 506–524, 2025 (in Chinese).

[15] G. Daras, C. A. Diaconu, E. Bagdasaryan, G. Frangella, and A. G. Dimakis, "A survey on diffusion models for inverse problems," arXiv preprint arXiv:2410.00083, 2024.

[16] Y. Li, K. Zhou, W. X. Zhao, and J.-R. Wen, "Diffusion models for non-autoregressive text generation: A survey," Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), 2023.

[17] Y. Yang et al., "A survey on diffusion models for time series and spatio-temporal data," arXiv preprint arXiv:2404.18886, 2024.

[18] M. M. Ahsan, S. Raman, Y. Liu, and Z. Siddique, "A comprehensive survey on diffusion models and their applications," arXiv preprint arXiv:2408.10207, 2024.
[19] Diffusers Team, "Stable Diffusion pipelines," Hugging Face documentation (online), accessed 2025.
[20] Y.-Y. Yang, "Diffusion model for time series and spatio-temporal data: A curated list," GitHub repository, 2024–2025.
[21] A. Kazerouni, "Awesome diffusion models in medical imaging," GitHub repository, 2023–2025.
[22] stablediffusionapi.com, "Stable Diffusion API services," technical documentation (online), accessed 2025.
[23] Hugging Face Spaces, "Stable Diffusion WebUI and related web user interfaces," community applications (online), accessed 2025.
[24] Stability AI, "Stability's API platform: simplifying API discovery and accelerating integration," developer blog posts and documentation, 2022–2023.
[25] Journal of Web Engineering, "Guidelines for Authors" and "Aims and Scope," Rinton Press / River Publishers (online), accessed 2025.
[26] B. Wei, D. Ruthven, M. Lalmas, and J. M. Jose, "A survey of faceted search," Journal of Web Engineering, vol. 12, no. 1–2, pp. 41–64, 2013.
[27] B. Ojokoh and E. Adebisi, "A review of question answering systems," Journal of Web Engineering, vol. 17, no. 8, pp. 717–758, 2019.
[28] W. Hawkins, B. Mittelstadt, and C. Russell, "Deepfakes on demand," Proc. ACM Conf. Fairness, Accountability, and Transparency (FAccT '25), ACM, New York, NY, USA, 13 pp., 2025.
[29] Reuters, "Getty Images largely loses landmark UK lawsuit over AI image generator," news report, 2025.

Biographies

Jee-Woo Shin is currently an undergraduate student in the Department of Computer Engineering at Hoseo University, Asan, Republic of Korea, where he enrolled in 2023. His research interests include artificial intelligence, machine learning, and related computational technologies.

Chayapol Kamyod achieved his Ph.D. in wireless communication from the Center of TeleInFrastruktur at Aalborg University, Denmark, a significant milestone in his academic career. This was preceded by a master's in electrical engineering from The City College of New York and, earlier, bachelor and master degrees in telecommunication engineering and laser technology and photonics from Suranaree University of Technology, Thailand. Currently, he is a lecturer in the Computer Engineering program at Mae Fah Luang University, Thailand, where his research is focused on the resilience and reliability of computer networks, wireless sensor networks, and exploring the potentials of IoT applications.

Chung-Pyo Hong received his B.Sc. and M.Sc. degrees in computer science from Yonsei University, Seoul, Korea, in 2004 and 2006, respectively. In 2012, he received his Ph.D. degree in computer science from Yonsei University, Seoul, Korea. He is currently an associate professor of Computer Engineering at Hoseo University, Asan, Korea. His research interests include machine learning, explainable AI, and data science.