Progressive Latent Refinement for Deadline-Aware Generative AI over Wireless Channels

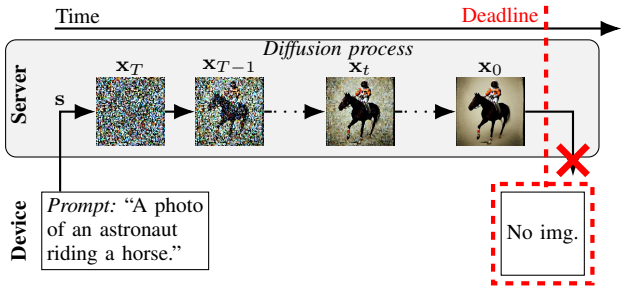
Anders E. Kalør and Tomoaki Ohtsuki
Department of Information and Computer Science, Keio University, Japan
E-mail: {aek, ohtsuki}@keio.jp

***Abstract*—Generative AI, such as diffusion models, faces challenges in real-time applications such as Extended Reality (XR) due to heavy computation and strict deadlines. Standard "compute-then-send" offloading often fails over dynamic wireless channels. We propose a progressive latent refinement strategy that leverages the iterative nature of diffusion models. The core of our solution is a tractable, analytical rate-distortion model, derived from the model's forward process, that predicts the final quality of any intermediate sample. This model drives a dynamic, online scheduling policy to intelligently selects which intermediate samples to transmit and at what rates. Simulation results show that our policy significantly outperforms baselines, delivering substantially higher sample quality at the deadline.**

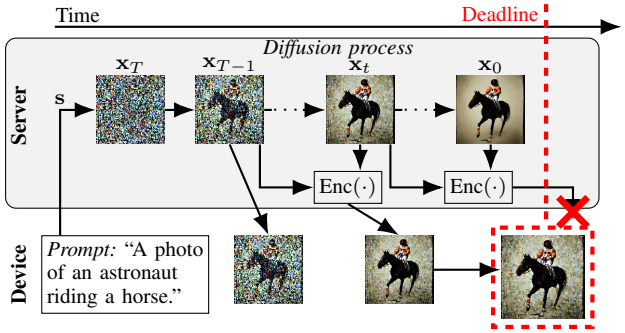
I. Introduction

Generative Artificial Intelligence (AI), and in particular diffusion models [1], [2], are expected to play a central role in emerging 6G applications, such as Extended Reality (XR), where they will support real-time content creation, rendering, and denoising [3]–[5]. Due to their heavy computational demands, diffusion models typically necessitate offloading their execution to edge servers. However, the long computation time of diffusion models combined with the stringent latency constraints of real-time interactive applications leaves little time for communicating the initial request (e.g., a prompt) and the final result. This poses a significant challenge for practical deployments, where the communication delay is subject to dynamic fluctuations in the wireless channel.

The majority of existing work on real-time edge AI has focused on efficient inference tasks, such as optimizing uplink transmission for split inference or semantic communication [5]–[7]. Works addressing the offloading of generative models, conversely, have primarily adopted a "compute-then-send" approach, where the full, high-fidelity output is generated at the edge before being transmitted [3], [5], [8]. This sequential process, however, is highly susceptible to network fluctuations and ill-suited for interactive applications, as a single, momentary deep channel fade can cause a missed deadline. Other approaches use diffusion to denoise channel noise, but do not address scheduling [9]. Separately, the concept of successive refinement and progressive transmission [10] has been applied to edge AI to adaptively balance latency, rate, and perception under time constraints. These strategies ensure that an intermediate result arrives by the deadline, even in poor channel conditions. However, the application of progressive techniques has been largely limited to the uplink feature transmission problem in split inference [11] or to general computational tasks [12].



(a) Traditional sequential generation and transmission



(b) Proposed progressive generation and transmission

Fig. 1. Sequential vs. progressive strategies for generative AI. (a): The traditional "compute-then-send" approach risks missing the deadline. (b): Our progressive scheme enables parallel communication and computation by transmitting delta-encoded intermediate results while the server generates further refinements. This ensures that a useful, intermediate result (in this case $\mathbf{x}_t$) is delivered on time.

In this paper, we propose a progressive refinement strategy for diffusion models that leverages their iterative structure (see Fig. 1). We develop two key components: 1) a novel *delta-encoding* scheme that exploits inter-sample correlation for aggressive, low-distortion quantization, and 2) a tractable online decision policy to select which samples to transmit. The policy's core is a novel, fully analytical rate-distortion model, derived from the diffusion model's forward process, that predicts the final distortion for any intermediate iteration and quantization rate. This model allows the policy to minimize the expected distortion at the deadline by dynamically balancing immediate transmission against waiting for future refinements. Simulations demonstrate significant gains in perceptual quality at the deadline compared to standard baselines.

II. System Model

A. Communication and Timing Model

We consider a deadline-constrained generative AI system where a mobile device (e.g., running an XR application) is connected to an edge server via a wireless link. The server's

task is to generate and communicate back to the device a latent sample $\mathbf{x}_0 \in \mathbb{R}^M \sim p(\mathbf{x}_0|\mathbf{s})$ based on an input $\mathbf{s}$ from the device, such as a text prompt for an image generation task. This latent sample can then be transformed into a final representation, such as a high-resolution image, using a lightweight decoder at the device, such as the decoder of a Variational Autoencoder (VAE) [2]. The server employs a T-step Latent Diffusion Model (LDM) to iteratively generate the sample $\mathbf{x}_0$ from initial noise $\mathbf{x}_T$, as detailed in Section III. We assume that time is divided into discrete slots, where one slot has a duration equal to the execution time of a diffusion step. The system operates under a strict deadline of T_d slots, by which a final, usable latent vector must be available at the device. We assume that the target latent samples $\mathbf{x}_0$ are normalized, i.e., $(1/M)\mathbb{E}[\|\mathbf{x}_0\|_2^2] = 1$, which is common in practical diffusion models.

The wireless link is modeled as a K-subchannel Rayleigh block-fading channel, with each sub-channel comprising ℓ symbols per slot. The total number of bits that can be communicated in slot n, denoted R_n, is given as

$$R_n = \ell \sum_{j=1}^{K} \log_2 \left(1 + \frac{p_{n,j} g_{n,j}}{N_0 W_\mathrm{s}}\right),$$

where W_s is the subchannel bandwidth, $p_{n,j}$ and $g_{n,j}$ are the transmission power and channel gain of subchannel j in slot n, respectively, and N_0 is the noise power spectral density. The channel gains are independent and identically distributed as $g_{n,j} \sim \mathrm{Exp}(1/\Gamma)$, where Γ is the average channel gain.

We assume perfect channel state information at the transmitter, allowing for optimal power control via waterfilling to maximize the rate [13]. Given the total power constraint P, the optimal power allocated to each subchannel is

$$p_{n,j} = \left(\frac{1}{\lambda} - \frac{N_0 W_\mathrm{s}}{g_{n,j}}\right)^+,$$

where $(x)^+ = \max(0, x)$ and λ is the Lagrange multiplier chosen to satisfy the total power constraint $\sum_{j=1}^{K} p_{n,j} = P$.

The input $\mathbf{s}$ transmitted in the uplink has a fixed size of B_s bits and occupies the first T_ul slots, given as

$$T_\mathrm{ul} = \min\left\{N_\mathrm{ul} \in \mathbb{N} : \textstyle\sum_{i=1}^{N_\mathrm{ul}} R_i \geq B_\mathrm{s}\right\}.$$

Upon receiving $\mathbf{s}$, the server begins the diffusion process in slot $T_\mathrm{ul} + 1$. Since each diffusion step takes one slot, the intermediate sample $\mathbf{x}_t$ becomes available in the beginning of slot $T_\mathrm{ul} + T - t + 1$. We denote the index of the most recent sample at the server at the beginning of slot n by $t(n)$, i.e.,

$$t(n) = T - \min\left(T, (n - T_\mathrm{ul} - 1)^+\right).$$

To maximize the quality of the sample at the device at the deadline, the server can progressively transmit intermediate samples $\mathbf{x}_t$ to the device before the final sample $\mathbf{x}_0$ is fully generated. To enhance compression, intermediate samples are delta-encoded relative to the most recent sample successfully reconstructed at the device (detailed in Section III). The number of slots required to transmit sample $\mathbf{x}_t$, encoded into B bits, starting from slot n is

$$T_\mathrm{dl}(n, B) = \min\left\{N_\mathrm{dl} \in \mathbb{N} : \textstyle\sum_{i=1}^{N_\mathrm{dl}} R_{n+i-1} \geq B\right\}.$$

B. Decision Problem Formulation

We formulate the problem of deciding which intermediate samples to transmit and at what rate as an online scheduling problem. The server executes a policy π at the beginning of each time slot $n \in \{1, \ldots, T_\mathrm{d}\}$. The possible actions are:

- $a_n = \textsc{transmit}(b)$: Transmit the most recently generated sample $\mathbf{x}_{t(n)}$ quantized to $b \in \mathcal{B}$ bits per dimension, where $\mathcal{B}$ is a set of predefined quantization rates. The resulting total transmission size is then $B = Mb$ bits (we neglect the overhead required to encode the choice of b). This action is only available if the diffusion model has started ($t(n) < T$) and if the channel is idle. The transmission will occupy the channel for $T_\mathrm{dl}(n, B)$ slots.
- $a_n = \textsc{wait}$: Do not transmit and wait for the next slot.

If a transmission is ongoing at the beginning of slot n or $t(n) = T$, the only possible action is $a_n = \textsc{wait}$.

Since the $\textsc{transmit}(b)$ action involves quantization, we will denote by $\hat{\mathbf{x}}_t$ the reconstructed version of the latent vector $\mathbf{x}_t$. Furthermore, let $\hat{\mathbf{y}}_n \in \{\hat{\mathbf{x}}_t\}_{t=0}^{T}$ denote the most recent sample successfully decoded by the device by the end of slot n. We assume the device and server have shared randomness, so the initial noise state $\mathbf{x}_T$ is identical, and the reconstructed version is lossless, i.e., $\hat{\mathbf{x}}_T = \mathbf{x}_T$. The device's initial state is thus $\hat{\mathbf{y}}_0 = \hat{\mathbf{x}}_T$, and this remains the state until the first downlink transmission is successfully decoded.

The objective of the policy π is to choose the sequence of actions $\{a_n\}_{n=1}^{T_\mathrm{d}}$ that minimizes the expected mean squared error (MSE) distortion between the latent sample available at the device at the deadline T_d, denoted $\hat{\mathbf{y}}_{T_\mathrm{d}}$, and the final, clean sample $\mathbf{x}_0$ that would be available with infinite time and bandwidth. This can be formalized as:

$$\min_{\pi} \quad (1/M)\mathbb{E}\left[\|\hat{\mathbf{y}}_{T_\mathrm{d}} - \mathbf{x}_0\|_2^2\right], \tag{1}$$

where the expectation is taken over the random channel realizations, the diffusion process, and the input data distribution for $\mathbf{s}$. Note that, by minimizing the MSE in the latent space, we are directly optimizing for a quantity intended to map to high quality of the final, decoded sample, such as the perceptual quality of a generated image.

III. Latent Refinement Coding for Diffusion Models

In this section, we present the proposed latent refinement coding scheme for diffusion models. We first provide a brief summary of LDMs and then present the coding scheme.

A. Latent Diffusion Models

An LDM generates a sample $\mathbf{x}_0$ from a distribution $p(\mathbf{x}_0|\mathbf{s})$ by reversing a gradual noising process, and consists of a fixed forward process and a learned reverse process [1], [2].

The forward process is defined as a discrete Markov chain

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\boldsymbol{\varepsilon}_{t-1},$$

for $t = 1, 2, \ldots, T$, where $\boldsymbol{\varepsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and $\alpha_t \in (0, 1)$ is a constant set by a noise schedule. This process gradually transforms any $\mathbf{x}_0$ into noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as $T \to \infty$. A central property is that $\mathbf{x}_t$ can be expressed in terms of $\mathbf{x}_0$ [1]:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\varepsilon}_t, \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. As assumed in the system model, $\mathbf{x}_0$ is normalized, and thus $(1/M)\mathbb{E}[\|\mathbf{x}_t\|_2^2] = 1$ for all t.

The reverse process $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$ iteratively denoises a sample $\mathbf{x}_T$ to generate $\mathbf{x}_0$. As the true reverse posterior is intractable, it is approximated by a learned process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$. This is typically accomplished by training a neural network $\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t, \mathbf{s})$ to predict the noise $\boldsymbol{\varepsilon}_t$ from $\mathbf{x}_t$ in Eq. (2). The reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$ is then simulated as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t, \mathbf{s})\right) + \sqrt{\tilde{\beta}_t}\boldsymbol{\varepsilon}', \tag{3}$$

where $\boldsymbol{\varepsilon}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}(1-\alpha_t)$. This allows the model to generate a new clean sample $\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{s})$ by recursively sampling $\mathbf{x}_{T-1}, \ldots, \mathbf{x}_0$ starting from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

B. Latent Refinement Coding of Diffusion Samples

To minimize expected distortion at the deadline, we propose to progressively transmit intermediate latent vectors $\mathbf{x}_t$. This strategy ensures a usable sample is delivered by the deadline even if poor channel conditions preclude transmitting the final $\mathbf{x}_0$, while also allowing gradual perceptual improvement for interactive applications.

Transmitting the full latent vector $\mathbf{x}_t$ at each update is highly inefficient as the iterative generation process creates strongly correlated samples. Instead, we propose a predictive coding scheme that transmits only the *refinement* (or "delta"). This refinement is computed as the difference between the current sample $\mathbf{x}_t$ and the previously reconstructed sample $\hat{\mathbf{x}}_{t_\mathrm{p}}$, which is available at both the server and device:

$$\boldsymbol{\delta}_{t,t_\mathrm{p}} = \mathbf{x}_t - \hat{\mathbf{x}}_{t_\mathrm{p}}. \tag{4}$$

the refinement is then quantized to b_t bits per dimension, $\hat{\boldsymbol{\delta}}_{t,t_\mathrm{p}} = Q_{b_t}(\boldsymbol{\delta}_{t,t_\mathrm{p}})$, and transmitted to the device. Given the quantized refinement $\hat{\boldsymbol{\delta}}_{t,t_\mathrm{p}}$, the device can reconstruct the next latent vector $\hat{\mathbf{x}}_t$ from its current one, $\hat{\mathbf{x}}_{t_\mathrm{p}}$, as

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t_\mathrm{p}} + \hat{\boldsymbol{\delta}}_{t,t_\mathrm{p}}. \tag{5}$$

Since the raw refinements $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ are generally small, they can be compressed significantly while keeping the distortion low.

The central problem is to construct a quantizer $Q_{b_t}(\cdot)$ for the refinements $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ that achieves the best rate-distortion tradeoff. However, rigorous analysis of the statistics of $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ is intractable due to its dependency on the quantized version of the previous sample, $\hat{\mathbf{x}}_{t_\mathrm{p}}$. Thus, to proceed we assume the quantization error is small and negligible, i.e., $\hat{\mathbf{x}}_{t_\mathrm{p}} \approx \mathbf{x}_{t_\mathrm{p}}$. This assumption, which we validate in Section V, allows us to model the statistics of the current refinement independently of potential error propagation from previous refinements. Unfortunately, the statistical properties of the approximated refinement $\boldsymbol{\delta}_{t,t_\mathrm{p}} \approx \mathbf{x}_t - \mathbf{x}_{t_\mathrm{p}}$ remain analytically intractable, as its distribution depends on $t_\mathrm{p} - t$ reverse diffusion steps, which are computed using the neural network $\boldsymbol{\varepsilon}_\theta(\cdot)$.

To create a tractable model, we leverage the insight that the learned reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$ is trained to approximate the true (but intractable) reverse posterior. We therefore propose to use the analytical properties of the *forward* process as a tractable approximation to model the statistics of the refinements generated by the *reverse* process.

To do this, we leverage the property of the forward process that for any $t_\mathrm{p} > t$, $\mathbf{x}_{t_\mathrm{p}}$ can be expressed as a noised version of $\mathbf{x}_t$ [1]:

$$\mathbf{x}_{t_\mathrm{p}} = \frac{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_t}}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_t}}\boldsymbol{\varepsilon}',$$

where $\boldsymbol{\varepsilon}' \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. By rearranging, we obtain an exact expression for $\mathbf{x}_t$ in terms of $\mathbf{x}_{t_\mathrm{p}}$ within the forward process:

$$\mathbf{x}_t = \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\mathbf{x}_{t_\mathrm{p}} - \frac{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\boldsymbol{\varepsilon}'. \tag{6}$$

We now use this ideal relationship as a tractable approximation for the statistics of the actual learned reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$. Equation (6) suggests that the refinement $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ can be approximated as

$$\begin{aligned}\boldsymbol{\delta}_{t,t_\mathrm{p}} &\approx \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\mathbf{x}_{t_\mathrm{p}} - \frac{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\boldsymbol{\varepsilon}' - \mathbf{x}_{t_\mathrm{p}} \\ &= \left(\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}} - 1\right)\mathbf{x}_{t_\mathrm{p}} - \frac{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\boldsymbol{\varepsilon}',\end{aligned}$$

where the approximation becomes an equality if the learned reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$ perfectly matches the true (but intractable) reverse posterior $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{s})$. It follows that the conditional distribution of the refinement $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ given $\mathbf{x}_{t_\mathrm{p}}$ is approximately Gaussian with mean $\boldsymbol{\mu}_{t,t_\mathrm{p}}$ and covariance $\sigma_{t,t_\mathrm{p}}^2\boldsymbol{I}$, where

$$\boldsymbol{\mu}_{t,t_\mathrm{p}} = \left(\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}} - 1\right)\mathbf{x}_{t_\mathrm{p}}, \qquad \sigma_{t,t_\mathrm{p}}^2 = \frac{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}{\bar{\alpha}_{t_\mathrm{p}}}. \tag{7}$$

This allows us to define a *standardized refinement* $\mathbf{z}_{t,t_\mathrm{p}}$ as

$$\begin{aligned}\mathbf{z}_{t,t_\mathrm{p}} &= \frac{\boldsymbol{\delta}_{t,t_\mathrm{p}} - \boldsymbol{\mu}_{t,t_\mathrm{p}}}{\sigma_{t,t_\mathrm{p}}} \\ &\approx \frac{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}\left((\mathbf{x}_t - \mathbf{x}_{t_\mathrm{p}}) - \left(\frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}} - 1\right)\mathbf{x}_{t_\mathrm{p}}\right) \\ &= \frac{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}{\sqrt{\bar{\alpha}_t - \bar{\alpha}_{t_\mathrm{p}}}}\left(\mathbf{x}_t - \frac{\sqrt{\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_{t_\mathrm{p}}}}\mathbf{x}_{t_\mathrm{p}}\right).\end{aligned}$$

Inserting our forward process approximation for $\mathbf{x}_t$ from (6) yields $\mathbf{z}_{t,t_\mathrm{p}} \approx -\boldsymbol{\varepsilon}'$. Thus, under our model, the complex, state-dependent refinement $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ can be transformed into a simple, state-independent Gaussian variable, $\mathbf{z}_{t,t_\mathrm{p}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Based on this result, we define the quantizer $Q_{b_t}(\cdot)$ as a function of a base scalar quantizer $Q_{\text{base},b_t}(\cdot)$:

$$Q_{b_t}(\boldsymbol{\delta}_{t,t_\mathrm{p}}) = \sigma_{t,t_\mathrm{p}} Q_{\text{base},b_t}\left(\frac{\boldsymbol{\delta}_{t,t_\mathrm{p}} - \hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}}}{\sigma_{t,t_\mathrm{p}}}\right) + \hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}},$$

where $\hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}} = \left(\sqrt{\bar{\alpha}_t/\bar{\alpha}_{t_\mathrm{p}}} - 1\right)\hat{\mathbf{x}}_{t_\mathrm{p}}$ is computed using the

reconstructed vector $\hat{\mathbf{x}}_{t_\mathrm{p}}$, and $Q_{\text{base},b_t}(\cdot)$ is a scalar quantizer designed for the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ using b_t bits per dimension, e.g., using Lloyd's algorithm [14]. We assume that base quantizers $Q_{\text{base},b}(\cdot)$ for each rate in $b \in \mathcal{B}$ are precomputed offline for a standard Gaussian distribution and shared between the server and device a priori.

The resulting algorithm can be implemented efficiently:

1) **Server:** Computes $\hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}}$, σ_{t,t_p}, and the normalized $\mathbf{z}_{t,t_\mathrm{p}} = (\boldsymbol{\delta}_{t,t_\mathrm{p}} - \hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}})/\sigma_{t,t_\mathrm{p}}$. It then applies the base quantizer $\hat{\mathbf{z}}_{t,t_\mathrm{p}} = Q_{\text{base},b_t}(\mathbf{z}_{t,t_\mathrm{p}})$ and transmits the bits for $\hat{\mathbf{z}}_{t,t_\mathrm{p}}$.
2) **Device:** Having $\hat{\mathbf{x}}_{t_\mathrm{p}}$, it re-computes the exact same $\hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}}$ and σ_{t,t_p}. Upon receiving $\hat{\mathbf{z}}_{t,t_\mathrm{p}}$, it performs the denormalization $\hat{\boldsymbol{\delta}}_{t,t_\mathrm{p}} = \sigma_{t,t_\mathrm{p}}\hat{\mathbf{z}}_{t,t_\mathrm{p}} + \hat{\boldsymbol{\mu}}_{t,t_\mathrm{p}}$.

This scheme simplifies the complex, state-dependent quantization of $\boldsymbol{\delta}_{t,t_\mathrm{p}}$ to the static quantization of a standard Gaussian variable, allowing a single codebook for $Q_{\text{base},b_t}(\cdot)$ to be used regardless of t and t_p.

IV. Dynamic Scheduling Policy

In this section, we develop a scheduling policy π to solve the distortion minimization problem in Eq. (1). Solving the problem optimally is computationally intractable due to the large state and action spaces. Instead, we develop a heuristic, online policy that relies on the analytical properties of the diffusion model. We first present a latent distortion model, a communication latency model, and finally formulate the proposed policy.

A. Latent Distortion Model

To guide the policy, the server must predict the distortion resulting from a given action. We define the total distortion $D_{\text{total}}(t, t_\mathrm{p}, b)$ as the MSE between the reconstructed $\hat{\mathbf{x}}_t$ and the final clean sample $\mathbf{x}_0$, i.e.,

$$D_{\text{total}}(t, t_\mathrm{p}, b) = \frac{1}{M}\mathbb{E}\left[\|\hat{\mathbf{x}}_t - \mathbf{x}_0\|_2^2 \,\middle|\, t_\mathrm{p}, b\right].$$

We approximate this as the sum of the diffusion distortion $D_{\text{diff}}(t) = (1/M)\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}_0\|_2^2]$ and the quantization distortion $D_{\text{quant}}(t, t_\mathrm{p}, b) = (1/M)\mathbb{E}[\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2]$, assuming that the two errors are uncorrelated:

$$D_{\text{total}}(t, t_\mathrm{p}, b) \approx D_{\text{diff}}(t) + D_{\text{quant}}(t, t_\mathrm{p}, b). \tag{8}$$

We first model the inherent diffusion distortion $D_{\text{diff}}(t)$ by again taking advantage of the tractable forward process. From Eq. (2), we have $\mathbf{x}_t - \mathbf{x}_0 = (\sqrt{\bar{\alpha}_t} - 1)\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t$. Since $\mathbf{x}_0$ and $\boldsymbol{\varepsilon}_t$ are uncorrelated and $\boldsymbol{\varepsilon}_t$ has zero mean, the distortion is

$$\begin{aligned} D_{\text{diff}}(t) &\approx (1/M)\mathbb{E}[\|(\sqrt{\bar{\alpha}_t} - 1)\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t\|_2^2] \\ &= \frac{1}{M}\left((\sqrt{\bar{\alpha}_t} - 1)^2\mathbb{E}[\|\mathbf{x}_0\|_2^2] + (1-\bar{\alpha}_t)\mathbb{E}[\|\boldsymbol{\varepsilon}_t\|_2^2]\right). \end{aligned}$$

Using the assumption that $(1/M)\mathbb{E}[\|\mathbf{x}_0\|_2^2] = 1$ and $(1/M)\mathbb{E}[\|\boldsymbol{\varepsilon}_t\|_2^2] = 1$, this simplifies to:

$$\begin{aligned} D_{\text{diff}}(t) &\approx (\bar{\alpha}_t - 2\sqrt{\bar{\alpha}_t} + 1) + (1 - \bar{\alpha}_t) \\ &= 2(1 - \sqrt{\bar{\alpha}_t}). \end{aligned} \tag{9}$$

Next, we model the quantization distortion $D_{\text{quant}}(t, t_\mathrm{p}, b)$. From Eqs. (4) and (5), the quantization error is simply the error in quantizing the refinement

$$\begin{aligned} \hat{\mathbf{x}}_t - \mathbf{x}_t &= \hat{\mathbf{x}}_{t_\mathrm{p}} + Q_b(\boldsymbol{\delta}_{t,t_\mathrm{p}}) - (\hat{\mathbf{x}}_{t_\mathrm{p}} + \boldsymbol{\delta}_{t,t_\mathrm{p}}) \\ &= Q_b(\boldsymbol{\delta}_{t,t_\mathrm{p}}) - \boldsymbol{\delta}_{t,t_\mathrm{p}}. \end{aligned}$$

Thus, the quantization distortion is given as

$$\begin{aligned} D_{\text{quant}}(t, t_\mathrm{p}, b) &= (1/M)\mathbb{E}[\|Q_b(\boldsymbol{\delta}_{t,t_\mathrm{p}}) - \boldsymbol{\delta}_{t,t_\mathrm{p}}\|_2^2] \\ &= (1/M)\sigma_{t,t_\mathrm{p}}^2\mathbb{E}[\|Q_{\text{base},b}(\mathbf{z}_{t,t_\mathrm{p}}) - \mathbf{z}_{t,t_\mathrm{p}}\|_2^2] \\ &= \sigma_{t,t_\mathrm{p}}^2 D_{\text{base}}(b), \end{aligned} \tag{10}$$

where $D_{\text{base}}(b) = \mathbb{E}[(Z - Q_{\text{base},b}(Z))^2]$ is the per-dimension distortion for $Z \sim \mathcal{N}(0,1)$. This quantity can be approximated in the high-rate regime using the Panter-Dite formula [15]:

$$D_{\text{base}}(b) \approx 2^{-2b}\frac{\pi\sqrt{3}}{2}.$$

For the initial transmission ($t_\mathrm{p} = T$) from $\hat{\mathbf{x}}_T$, we define $\sigma_{t,t_\mathrm{p}}^2 = 1$, yielding $D_{\text{quant}}(t, T, b) \approx D_{\text{base}}(b)$, while for refinements we use the expression from Eq. (7). If no transmission has yet occurred, the device holds $\hat{\mathbf{x}}_T = \mathbf{x}_T$, and the total distortion is $D_{\text{total}}(T, T, \infty) = D_{\text{diff}}(T)$.

B. Communication Latency Model

Besides predicting the distortion, the policy must also be able to predict the latency cost of a TRANSMIT(b) action. The number of slots $T_{\text{dl}}(n, B)$ required to transmit B bits is a random variable, depending on future channel realizations. A simple point estimate $\hat{T} = B/\mathbb{E}[R_n]$ is insufficient for a deadline-constrained system, as it provides no guarantee of completion. Instead, we model the q-th quantile of the latency, T_q, representing the time by which the transmission will be complete with probability q.

Since the number of subchannels K is large, by the central limit theorem, the rate R_n in any given slot is approximately Gaussian, $R_n \sim \mathcal{N}(\bar{R}, \sigma_R^2)$. The total bits supported by N slots, $S_N = \sum_{i=1}^N R_i$, is thus also approximately Gaussian:

$$S_N \sim \mathcal{N}(N\bar{R}, N\sigma_R^2).$$

The quantile latency T_q can thus be obtained by solving $\Pr(S_{T_q} \geq B) = q$, which is equivalent to

$$1 - \Phi\left(\frac{B - T_q\bar{R}}{\sqrt{T_q}\sigma_R}\right) = q,$$

where $\Phi(\cdot)$ is the Gaussian cumulative distribution function. By rearranging and substituting $x = \sqrt{T_q}$, T_q can be estimated by solving the following quadratic equation for x:

$$\bar{R}x^2 + \left(\Phi^{-1}(1-q)\sigma_R\right)x - B = 0, \tag{11}$$

and setting $\hat{T}_q = \lceil x^2 \rceil$.

To find the ergodic rate $\bar{R} = \mathbb{E}[R_n]$ and variance σ_R^2, we use the assumption that K is large to replace empirical averages over subcarriers with statistical expectations [13]. First, we find the waterfilling cutoff level $\gamma_0 = \lambda N_0 W_s$ by numerically solving the total power constraint $P = K\mathbb{E}_g\left[\left(\frac{N_0 W_s}{\gamma_0} - \frac{N_0 W_s}{g}\right)^+\right]$ given the Rayleigh distribution

$g \sim \text{Exp}(1/\Gamma)$. Given the solution γ_0, the ergodic rate $\bar{R}$ is computed by integrating the channel capacity

$$\bar{R} = \ell K \mathbb{E}_g \left[\log_2 \left(1 + \frac{1}{N_0 W_s} \left(\frac{1}{\lambda} - \frac{N_0 W_s}{g} \right)^+ g \right) \right] = \frac{\ell K}{\Gamma} \int_{\gamma_0}^{\infty} \log_2 \left(\frac{g}{\gamma_0} \right) e^{-(1/\Gamma)g} dg.$$

The variance $\sigma_R^2 = \mathbb{E}[R_n^2] - \bar{R}^2$ can be found similarly by numerically computing $\mathbb{E}[R_n^2]$.

C. Policy Formulation

We now formulate the scheduling policy based on the distortion and latency models. An optimal policy is computationally intractable because the value of any action depends on all future states, and instead we therefore develop a heuristic, online policy. However, since a WAIT action provides no immediate improvement, a simple myopic policy of just comparing the TRANSMIT(b) action to WAIT is insufficient as it would always favor transmission. The value of waiting is to allow the server to compute a more refined sample for a later transmission. Our policy models this non-trivial tradeoff. At any slot n where $t(n) < T$ and the channel is free, our policy performs a lookahead to compare the two dominant strategies of transmitting either once or twice before the deadline.

Let $D_{\text{total}}(t, t_{\text{p}}, b)$ be the estimated distortion from Eq. (8) and $\hat{T}_q(B)$ be the approximated latency quantile from Eq. (11) for $B = Mb$ bits. Let t_{p} denote the index of the most recent sample successfully decoded by the device (e.g., $t_{\text{p}} = T$ at the beginning). We treat the quantile q as a fixed hyperparameter. The two evaluated strategies are:

1) **Strategy 1 (Optimal Single Transmission):** This strategy models the "wait-then-send" approach. By searching all possible wait times $\tau \in \{0, 1, \ldots, T_{\text{d}} - n\}$ (where $\tau = 0$ means transmit now), it finds the best possible distortion D_1^* achievable by transmitting a single sample $\mathbf{x}_{t(n+\tau)}$ with quantization rate $b_\tau^{\max}$ before the deadline. Specifically, it computes
$$D_1^* = \min_{\tau \in \{0, \ldots, T_{\text{d}} - n\}} D_{\text{total}}(t(n+\tau), t_{\text{p}}, b_\tau^{\max}),$$
where $b_\tau^{\max} \in \mathcal{B}$ is the maximum feasible quantization rate such that $n + \tau + \hat{T}_q(Mb_\tau^{\max}) \le T_{\text{d}}$.
2) **Strategy 2 (Optimal Dual Transmission):** This strategy models the "send-and-refine" progressive approach. It computes the best possible distortion D_2^* achievable by transmitting the current sample $\mathbf{x}_{t(n)}$ immediately with quantization rate b, followed by a transmission of a future refinement at time $n + \hat{T}_q(Mb) + \tau'$:
$$D_2^* = \min_{b \in \mathcal{B}} \min_{\tau'} D_{\text{total}}(t(n + \hat{T}_q(Mb) + \tau'), t(n), b_{\tau'}^{\max}),$$
where $\tau' \in \{0, \ldots, T_{\text{d}} - n - \hat{T}_q(Mb)\}$ and $b_{\tau'}^{\max} \in \mathcal{B}$ is the maximum quantization rate satisfying $n + \hat{T}_q(Mb) + \tau' + \hat{T}_q(Mb_{\tau'}^{\max}) \le T_{\text{d}}$.

Let τ^* denote the optimal wait time from Strategy 1 and b^* the optimal quantization rate from Strategy 2. If $D_2^* \le D_1^*$, the server executes TRANSMIT(b^*). If $D_1^* < D_2^*$ and $\tau^* = 0$, the policy executes action TRANSMIT($b_0^{\max}$); otherwise, if $D_1^* < D_2^*$ and $\tau^* > 0$, it executes WAIT. This policy captures the central tradeoff between transmitting immediately versus waiting for higher-quality refinements, and is computationally tractable as its overhead is negligible compared to a diffusion step, since it only requires searching over a small, one-dimensional space of wait times and quantization levels.

V. Numerical Results

A. Experimental Setup

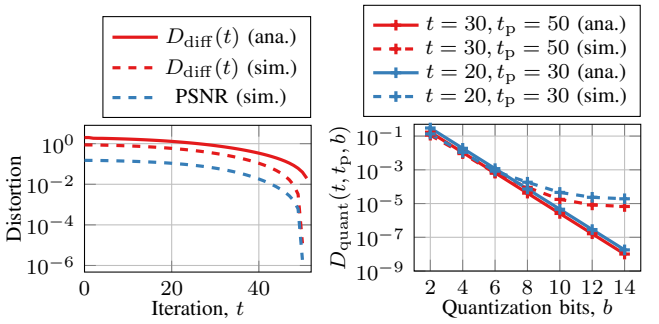
We evaluate our policy using a pre-trained Stable Diffusion v1-5 model [2] that generates images based on an input text prompt. This model operates on an $M = 64 \times 64 \times 4$-dimensional latent space, with a VAE decoder to map the latent vector to a $512 \times 512 \times 3$-dimensional image. The reverse process uses $T = 50$ steps, and we set a strict system deadline of $T_{\text{d}} = 60$ time slots. The uplink prompt size is fixed to $B_{\text{s}} = 8192$ bits. The wireless link consists of $K = 60$ subchannels, each with $\ell = 70$ downlink symbols/slot (corresponding to approximately 5 ms in a typical 5G setting with a symmetric uplink/downlink split), and we evaluate across a range of average channel SNRs, defined as $\text{SNR} = P\Gamma/(N_0 W_s)$. The policy's latency model Eq. (11) uses a $q = 0.9$ quantile target, and we fix the set of possible quantization levels to $\mathcal{B} = \{3, 4, 8, 10, 16\}$, where 16 bits/dimension corresponds to lossless (i.e., $D_{\text{quant}}(t, t_{\text{p}}, 16) = 0$). We generate the results using 500 prompts drawn from the DiffusionDB dataset [16].

The quality of the sample delivered at the deadline is quantified using two metrics: latent MSE $(1/M)\mathbb{E}[\|\hat{\mathbf{y}}_{T_{\text{d}}} - \mathbf{x}_0\|_2^2]$, which our policy aims to minimize (Eq. (1)), and the *LPIPS* score [17], which measures the perceptual quality of the decoded image. We compare our policy against two baselines:

- **Persistent-Q**: A naive progressive scheme that transmits every new sample $\mathbf{x}_t$ (or as fast as the channel allows) using a fixed bitrate of b_{fix} bits/dim.
- **Final-Only**: Waits until the server computes $\mathbf{x}_0$ (at slot $T_{\text{ul}} + T + 1$). It then transmits $\mathbf{x}_0$ using the lossless $b_0 =$ 16-bit representation.

B. Performance Evaluation

We first validate the accuracy of our latent distortion models, which are the fundamental components of our policy. Figure 2 shows the predicted distortions from Eqs. (9) and (10) against the simulated results. As shown in Fig. 2a, our $D_{\text{diff}}(t)$ model captures the decreasing trend of the true, simulated latent distortion, although with a slight offset. Furthermore, this figure also plots the final image's pixel-space distortion (as $1/\text{PSNR}$), which almost perfectly tracks the empirical latent distortion D_{diff}. This suggests that minimizing the latent-space error is a valid proxy for maximizing the final perceptual quality. The quantization distortion (Fig. 2b) accurately predicts the trend, especially at low-to-mid bitrates. The simulated results flatten at high bitrates (e.g., $b \ge 8$), which we attribute to our model's forward-process Gaussianity assumption not perfectly matching the learned reverse process. In both cases, the discrepancies are expected and highlight the

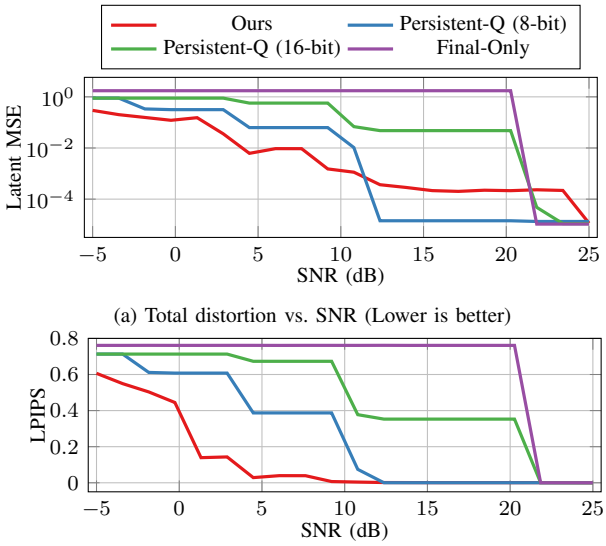


(a) Diffusion distortion (b) Quantization distortion

Fig. 2. Validation of the analytical distortion models.

approximation inherent in using the forward process to model the learned reverse process. Yet, as Fig. 3 will show, these models are highly effective for policy decisions.

Figure 3 shows the final sample quality at the device at the deadline versus the average channel SNR. Figure 3a plots the latent MSE (the objective our policy directly minimizes), while Fig. 3b shows the resulting perceptual quality. Our policy significantly outperforms all baselines in the low-to-mid SNR range. The 'Final-Only' baseline fails completely until the SNR reaches approximately 21 dB, as the few slots remaining after generation ($T_{\text{d}} - (T_{\text{ul}} + T)$) are insufficient for the large, lossless transmission. In contrast, our policy selects a lower-quality but transmissible sample, guaranteeing a usable result by the deadline and achieving significant gains. This demonstrates that our analytical distortion models are sufficiently accurate to guide the policy toward effective decisions. At high SNR, our policy results in a slightly larger latent MSE than 'Persistent-Q (8 bit)', which we attribute to approximation errors in our analytical framework and the simplified policy. However, its equivalent perceptual quality (LPIPS) confirms that this discrepancy is perceptually negligible. This combined performance across SNRs demonstrates our policy's adaptability to channel conditions and its effectiveness in using latent MSE as a proxy to optimize for perceptual quality, which is the key end-user metric for generative models.



(a) Total distortion vs. SNR (Lower is better)

(b) LPIPS vs. SNR (Lower is better)

Fig. 3. Final sample quality at the deadline ($T_{\text{d}} = 60$) versus channel SNR.

VI. Conclusion

We have proposed a novel framework for deadline-aware generative AI over wireless links, centered on a progressive latent refinement strategy. The core of our solution is a tractable, analytical rate-distortion model, derived from the diffusion model's forward process. This model enables a dynamic scheduling policy to predict the quality of intermediate samples without empirical characterization. Simulation results confirmed the accuracy of our rate-distortion model and demonstrated that our proposed policy significantly outperforms baselines in terms of both achieved distortion and perceptual quality at the deadline.

Acknowledgment

This work was supported by the Japan Science and Technology Agency (JST) ASPIRE program (grant no. JPMJAP2326) and by the Mizuho Foundation for the Promotion of Sciences.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neur. Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.
[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. CVPR*, June 2022, pp. 10 684–10 695.
[3] M. Xu *et al.*, "Unleashing the power of edge-cloud generative AI in mobile networks: A survey of AIGC services," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1127–1170, 2024.
[4] A. E. Kalør *et al.*, "Wireless 6G connectivity for massive number of devices and critical services," *Proc. IEEE*, pp. 1–23, 2024.
[5] Z. Liu *et al.*, "Integrated sensing and edge AI: Realizing intelligent perception in 6G," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2025.
[6] Z. Wang, A. E. Kalør, Y. Zhou, P. Popovski, and K. Huang, "Ultra-low-latency edge inference for distributed sensing," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2025.
[7] Q. Zeng *et al.*, "Knowledge-based ultra-low-latency semantic communications for robotic edge intelligence," *IEEE Trans. Commun.*, vol. 73, no. 7, pp. 4925–4940, 2025.
[8] H. Du *et al.*, "Diffusion-based reinforcement learning for edge-enabled AI-generated content services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8902–8918, 2024.
[9] L. Guo *et al.*, "Diffusion-driven semantic communication for generative models with bandwidth constraints," *IEEE Trans. Wireless Commun.*, vol. 24, no. 8, pp. 6490–6503, 2025.
[10] W. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 269–275, 1991.
[11] G. Zhang *et al.*, "Progressive learned image transmission for semantic communication using hierarchical vae," *IEEE Trans. Cogn. Commun. Netw.*, pp. 1–1, 2025.
[12] H. Esfahanizadeh, A. Cohen, S. S. Shitz, and M. Médard, "Successive refinement in large-scale computation: Expediting model inference applications," *IEEE Trans. Signal Process.*, 2025.
[13] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.
[14] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[15] P. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proc. IRE*, vol. 39, no. 1, pp. 44–48, 1951.
[16] Z. J. Wang *et al.*, "DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models," *arXiv:2210.14896*, 2022.
[17] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.