

Appendix

A Numerical Details

A.1 MPRL Upper Bound

This part delves into the derivation of Eq. (18). We first express the expected loss in terms of the optimal estimator (using shorthand notation subsequently):

$$E[l(X, \hat{X}(X, \gamma))] = E \left[X \log \frac{X}{\hat{X}(X, \gamma)} - X + \hat{X}(X, \gamma) \right] = E \left[X \log \frac{X}{\hat{X}^*} \right] + E \left[X \log \frac{\hat{X}^*}{\hat{X}} - X + \hat{X} \right]. \quad (20)$$

Using the law of iterated expectation gives:

$$E[l(X, \hat{X})] = \text{mprl}(\gamma) + E[l(\hat{X}^*, \hat{X})].$$

The second term above denotes the estimation gap, and rearranging the terms, we get:

$$E[-\log P(x)] = \int_0^\infty \left(E[l(X, \hat{X})] - E[l(\hat{X}^*, \hat{X})] \right) d\gamma.$$

Using Jensen’s inequality here (based on the properties mentioned in Lemma 1), we have:

$$-E[l(\hat{X}^*, \hat{X})] \leq -E[\hat{X}^*] \log \frac{E[\hat{X}^*]}{\hat{X}} - E[\hat{X}^*] + \hat{X} = -l(E[X], \hat{X}). \quad (21)$$

Now, using the relation from Lemma 2 gives us:

$$\int_0^\infty \text{mprl}(\gamma) d\gamma \leq \int_0^\infty E[l(X, E[X])] d\gamma. \quad (22)$$

We obtain a more elegant bound in terms of our suboptimal neural denoiser by dropping the negative term:

$$E[-\log P(x)] = \int_0^\infty \text{mprl}(\gamma) d\gamma \leq \int_0^\infty E[l(X, \hat{X})] d\gamma. \quad (23)$$

A.2 Numerical Integration.

This section outlines the effective computation of integral from (19). We first use importance sampling to rewrite the integral as an expectation over a distribution, $q(\gamma)$, allowing for unbiased Monte Carlo estimation. This leads to our final numerical approximation of the loss function $E_{p(x)}[-\log p(x)] \leq \mathcal{L}$, where

$$\mathcal{L} \equiv E_{q(\alpha)} \left[\frac{1}{q(\alpha)} E_{(x, z_\gamma)} [l(X, \hat{x})] \right].$$

We propose two paradigms for numerical integration: **Logistic** and **Uniform** Integration, respectively.

Logistic Integration. In Gaussian diffusion models, the log-SNR integral is approximated via importance sampling with a truncated logistic distribution. The integrand, shaped by a mixture of logistic CDFs influenced by data covariance eigenvalues λ_i , is captured by matching the empirical mean μ and variance s of $-\log \lambda_i$, with integration bounds $[\mu - 4s, \mu + 4s]$. Samples drawn via the logistic quantile function are weighted by $1/q(\alpha)$ to prioritize critical regions, reducing variance.

Uniform Integration. This simpler numerical method discretizes the log-SNR range $[\alpha_1, \alpha_2]$ into a uniform grid, applying trapezoidal or Riemann-sum integration without assuming an underlying distribution. While simple, efficiency depends on grid density for broad ranges, favoring ease over optimal sampling. The predefined range is $[-28, 37]$ with uniform sampling.

B Experimental Details

B.1 Training Details (contd.)

For a fair comparison, we train both CIFAR and LMD models from scratch for 600 epochs. The training starts with a learning rate of 2×10^{-5} using the Adam optimizer. We adopt an 80-20 train-test

split for evaluating likelihoods. For image generation, we use a UNet-based model[36], while for music generation, we employ the DenseDDPM[37] and convolutional-transformer[17]-based models for the continuous embeddings (DDPM-style) and discrete domain (D3PM[17]) respectively. The training procedure ensures consistency across both domains, facilitating a meaningful comparison of their performance. It is to be noted that we train all of the models from scratch, owing to a lack of pre-trained Poisson diffusion baselines, to ensure fair comparison. Because of compute resource constraints, we train the models upto 600 epochs, which falls short of the usual amount of training required to achieve peak performance (e.g., LTJ[20] trains their models for 3600 epochs). We also restrict ourselves to 100 logSNR values per image / music sample, and restrict the number of denoising steps used in the DDPM / D3PM baselines to 100 as well (instead of 1000), to ensure fair comparison. Thus, although the relative performance of the models is preserved, the absolute values of the metrics underperform those presented in DDPM[13] and LTJ[20].

B.2 Data and Model Normalization

We experimented with various schemes for data (D_n) before passing it through the noisy channel and for model inputs (M_n) post-noising. CIFAR-10 data is normalized to $[0, 1]$, $[1, 2]$, $[0, 255]$, $[-1, 1]$; Lakh MIDI to $[0, 1]$, $[1, 2]$, $[0, 90]$, $[-1, 1]$. Poisson channels cannot handle negatives and since zero inputs yield zeros, we shift inputs by $\epsilon = 10^{-6}$. For Gaussian noising, model normalization used $[0, 1]$ or $[-1, 1]$, while Poisson noising used only $[0, 1]$. The best results were achieved with $[-1, 1]$ (Gaussian) and $[1, 2]$ (Poisson) for D_n , and $[-1, 1]$ (Gaussian) and $[0, 1]$ (Poisson) for M_n . Among the integration paradigms used, logistic integrate yielded the best empirical results, and the `loc` and `scale` parameters obtained for the mid-integral range were (6, 3) for Gaussian noising and $(-1, 5)$ for Poisson noising.

B.3 Denoiser Architecture

For CIFAR-10 images, we employ a U-Net architecture [36] with residual blocks and self-attention layers. The encoder comprises four downsampling blocks (convolution \rightarrow GroupNorm \rightarrow SiLU) that reduce spatial resolution from 32×32 to 4×4 , followed by a bottleneck with self-attention at 8×8 resolution. The decoder mirrors the encoder via transposed convolutions and skip connections. For symbolic music synthesis on Lakh MIDI, we use a DenseDDPM[50]-based architecture and a convolutional transformer[19]-based model, for the continuous-state DDPM modeling and the discrete D3PM[19] modeling respectively. For the continuous modeling, we adapt the DenseDDPM architecture from [37]. It first projects the input latent vector to an MLP hidden size (default 2048) with a single Dense layer, then runs it through 3 residual MLP blocks whose weights are modulated by a 128-dimensional sinusoidal embedding of the diffusion timestep t . After these conditioned residual blocks, it applies a LayerNorm and a final Dense layer that maps back to the original latent dimensionality, yielding the denoised output. For the discrete modeling, we adapt an NCSN++ backbone [50] with a Convolutional Transformer encoder [19]. The architecture includes a 512-dimensional embedding layer, six transformer layers with multi-head attention (8 heads) and positional encodings, and time-dependent noise conditioning.

B.4 Symbolic Music Dataset Cleanup

We utilize the cleaned Lakh MIDI dataset [40], loading note sequences from .npy files with original shape $(x, 1024)$. For training, sequences are partitioned into individual 1D vectors of shape $(1, 1024)$, representing discrete musical events. So, our method directly models symbolic music as discrete 1D note sequences using Poisson diffusion, avoiding hybrid architectures or preprocessing.

B.5 Domain-Specific Metrics

To evaluate the generation quality of our model across image and audio domains, we utilize established domain-specific metrics that quantify fidelity, diversity, and structural realism. Below, we provide descriptions and implementation details for each metric employed in our evaluation.

Image Metrics All image-generation metrics were computed on 40,000 randomly selected ground-truth images from the CIFAR-10 test split and 40,000 model-generated samples. Fréchet Inception Distance (FID) was evaluated with the PyTorch `torch-fidelity` package (Inception-v3 network, 2048-dimensional pool3 activations).

- **Structural Similarity Index Measure (SSIM)** [41]: SSIM measures the similarity between two images by comparing their luminance, contrast, and structure. It is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ and σ denote mean and standard deviation over local image patches. Higher SSIM indicates better perceptual similarity.

- **Fréchet Inception Distance (FID)** [42]: FID evaluates the distance between real and generated image distributions in the feature space of a pretrained Inception network. It is calculated as:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the means and covariances of the feature embeddings of real and generated samples.

Audio Metrics. All audio-based metrics are computed using 10,000 ground-truth samples and 10,000 generated samples per model. To enable consistent audio evaluation, we first convert model-generated .npy files to MIDI format using the `pretty_midi` library. These MIDI files are then rendered to WAV audio using `FluidSynth` [53] with the `FluidR3_GM` soundfont, ensuring uniform timbre across all samples. All tools and dependencies are managed within an automated evaluation pipeline. This standardized conversion procedure ensures reproducibility and fair comparison of audio metrics across all models.

- **Fréchet Audio Distance (FAD)** [43]: Analogous to FID, FAD computes the Fréchet distance between embeddings of real and generated audio, extracted via a VGGish model pretrained for audio classification. It reflects perceptual similarity in the feature space and is calculated similarly to FID.
- **Consistency (C)**: To evaluate sequence-level realism, we employ framewise self-similarity based on overlapping Gaussian approximations of pitch histograms. Specifically, we use the overlapping area (OA) from [44], applied to pitch only (since duration is fixed in our setup). For sliding 4-measure windows with 2-measure hop:

$$\text{OA}(k, k+1) = 1 - \text{erf}\left(\frac{c - \mu_1}{\sqrt{2}\sigma_1}\right) + \text{erf}\left(\frac{c - \mu_2}{\sqrt{2}\sigma_2}\right)$$

The resulting pitch OA values are compared to ground-truth sequences via:

$$C = \max\left(0, 1 - \frac{|\mu_{\text{OA}} - \mu_{\text{GT}}|}{\mu_{\text{GT}}}\right)$$

$$\text{Var} = \max\left(0, 1 - \frac{|\sigma_{\text{OA}}^2 - \sigma_{\text{GT}}^2|}{\sigma_{\text{GT}}^2}\right)$$

Consistency (C) measures global similarity to ground truth, while variance (Var) captures generation diversity. High C implies structured, music-like pitch transitions.

- **Mel-Spectrogram Inception Distance (MSID)** [45]: MSID adapts FID for audio by computing the Fréchet distance over features extracted from Mel spectrograms. The key steps include:

- Convert generated .npy files to MIDI and synthesize audio using `FluidSynth`.
- Compute 128-band Mel spectrograms (16kHz, FFT=2048, hop=512), as outlined in B.5.
- Extract features using a VGG16-based architecture trained on audio (VGGish).
- Compute MSID using: $\text{MSID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$

MSID captures both spectral and perceptual differences, correlating with human audio quality judgments.

- **Wasserstein Distance (WD)** [54]: WD quantifies the distance between the token distributions of real and generated symbolic music. We compute a *weighted Wasserstein distance* that prioritizes important token types (e.g., binary onsets or active pitches):

$$W_w(p, q) = \inf_{\gamma \in \Pi(p, q)} \mathbb{E}_{(x, y) \sim \gamma} [c(x, y) \cdot w(x, y)]$$

Weights are assigned based on token values: 0.2 for 0s, 0.5 for 1s, 1.0 for others. Tokens are normalized and reshaped as needed. Lower WD values indicate better alignment of pitch activation distributions.

In addition to the core domain-specific metrics described in Appendix B.5, we include the following complementary metrics used for additional analysis presented in Table 4. These metrics help analyze fine-grained perceptual and structural properties of the generated data.

Images:

- **Learned Perceptual Image Patch Similarity (LPIPS)** [55]: LPIPS measures perceptual similarity by computing the distance between deep features extracted from pretrained vision networks (e.g., VGG, AlexNet). It is defined as:

$$\text{LPIPS}(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h, w} \|w_l \odot (\phi_l^x(h, w) - \phi_l^y(h, w))\|_2^2$$

where ϕ_l^x and ϕ_l^y are feature activations from layer l , and w_l are learned weights. Lower LPIPS values indicate higher perceptual similarity between generated and reference images.

Audio:

- **Spectral Convergence (SC)**: SC quantifies the relative difference between the magnitude spectra of real and generated audio:

$$\text{SC} = \frac{\| |S_{\text{gen}}| - |S_{\text{ref}}| \|_F}{\| |S_{\text{ref}}| \|_F}$$

where S_{gen} and S_{ref} are the STFTs (Short-Time Fourier Transforms) of generated and reference audio, and $\| \cdot \|_F$ denotes the Frobenius norm. Lower SC suggests higher spectral alignment.

- **Log Mean Spectral Distance (LMSD)**: LMSD captures differences in log-scaled spectral magnitudes and is defined as:

$$\text{LMSD} = \frac{1}{T} \sum_t \|\log(\epsilon + |S_{\text{gen}}(t)|) - \log(\epsilon + |S_{\text{ref}}(t)|)\|_1$$

where ϵ is a small constant to ensure numerical stability, and the summation is over time frames t . Lower LMSD implies improved perceptual quality in frequency response.

- **Variance (Pitch Histogram Diversity)**: [37] As described in Appendix B.5, we also compute the pitch variance metric (*Var*) to measure structural diversity in symbolic music:

$$\text{Var} = \max \left(0, 1 - \frac{|\sigma_{\text{OA}}^2 - \sigma_{\text{GT}}^2|}{\sigma_{\text{GT}}^2} \right)$$

Higher variance indicates greater distributional diversity while maintaining similarity to ground truth statistics. Together, these metrics offer a comprehensive, multi-faceted evaluation of image and audio generation quality, balancing fidelity, diversity, and perceptual structure.

Mel Spectrogram Computation Parameters:

For the listed audio-based metrics (FAD, MSID, SC, LMSD), we first convert generated symbolic music into waveform as discussed earlier [53] and compute Mel spectrograms with the following parameters:

- **Sampling rate:** 16 kHz — chosen to balance temporal resolution and frequency coverage for symbolic music.
 - **FFT size:** 2048 — defines the window size for frequency analysis. This size gives sufficient frequency granularity (≈ 7.8 Hz per bin at 16 kHz).
 - **Hop length:** 512 — determines the stride between successive STFT windows, corresponding to 32 ms hop (suitable for music temporal structure).
 - **Mel bands:** 128 — provides a perceptually motivated representation of frequency, emphasizing resolution in lower frequency ranges where musical structure is denser.
- These parameters are consistent with best practices in neural audio synthesis [56],[45] and ensure compatibility with pretrained perceptual models like VGGish.

Additional Metrics:

Table 4: Auxiliary generative quality metrics. Image: LPIPS; Audio: SC, LMSD, LPIPS (Mel), Var

Baseline	LPIPS (Img)	SC	LMSD	LPIPS (Mel)	Var
IDDPM [38]	0.17 ± 0.05	1.56	9.99	0.38 ± 0.10	0.81
LTJ [20]	0.18 ± 0.06	1.51	9.81	0.33 ± 0.10	0.87
D3PM [17]	0.29 ± 0.09	1.41	9.63	0.28 ± 0.09	0.90
ItDPDM	0.18 ± 0.08	1.49	9.71	0.30 ± 0.09	0.85

B.5.1 Visualizing generated music samples:

Individual ItDPDM Samples: To examine local model behavior, we present isolated piano roll visualizations of individual samples (see Figure 8). Each plot shows the temporal and pitch structure of a single sequence, with color indicating note velocity. These visualizations enable detailed inspection of rhythmic patterns, pitch range, note density, and artifacts.

For example, ItDPDM-generated samples exhibit consistent pitch contours and relatively uniform spacing, occasionally disrupted by outlier notes or sparse regions. Such plots help diagnose issues like over/under-generation, discontinuities, or anomalies, and complement the broader comparisons across models.

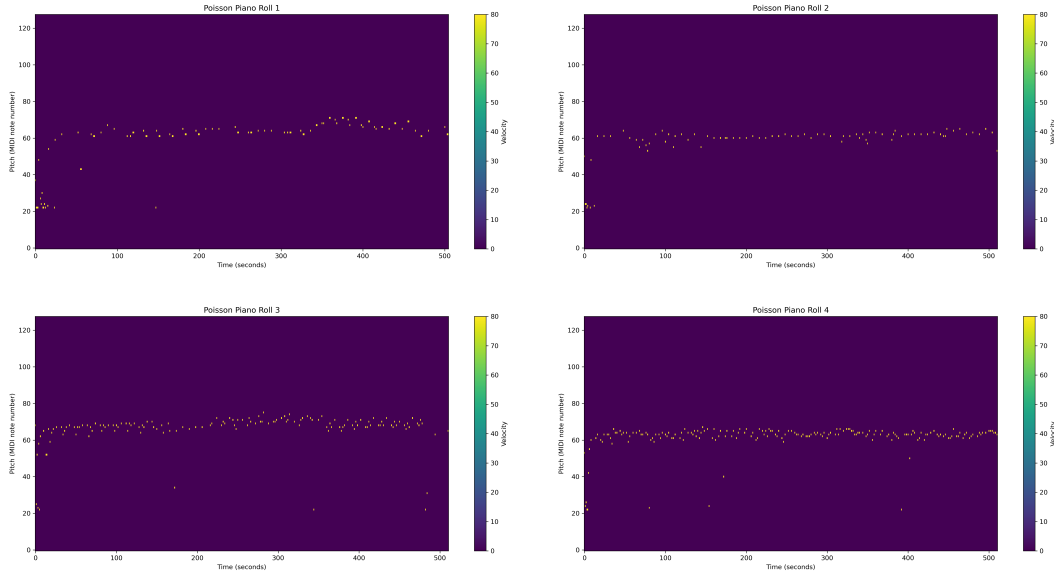


Figure 8: Isolated piano roll visualizations of four ItDPDM-generated samples. Each plot shows pitch over time, with note velocity indicated by color intensity.

Qualitative comparison: To qualitatively observe the generative performance of our models, we visualize representative samples as piano rolls in Figure 9. Each row presents a different generated

sequence, with columns corresponding to different models: DDPM (left), ASD3PM (center), and ItDPDM (right). Each piano roll plot depicts note pitch (vertical axis) over time (horizontal axis), with intensity indicating note onset.

DDPM (left): Samples from DDPM display high variability in pitch and rhythm, with note events appearing scattered and less structured. While diverse, these outputs often lack recognizable musical motifs or rhythmic regularity, indicating that the model struggles to capture long-range musical structure.

ASD3PM (center): ASD3PM outputs, derived from perturbed ground truth MIDI sequences, exhibit strong rhythmic and melodic coherence. These samples closely mirror the structure of real music, featuring sustained motifs, consistent phrasing, and regular timing. This visual consistency aligns with the model’s design, which prioritizes fidelity to the data manifold.

ItDPDM (right): Samples from ItDPDM demonstrate improved musical structure over DDPM. While some randomness remains, many outputs show rhythmic grouping, pitch contours, and repeating patterns, suggesting the model’s ability to learn and replicate fundamental elements of musical organization. Overall, the visualizations highlight key differences in generative behavior. ASD3PM achieves the highest structural fidelity, followed by ItDPDM, which balances diversity with coherence. DDPM produces varied outputs but lacks the structured rhythmic and melodic features observed in the other methods. These qualitative findings complement our quantitative results, offering insight into how each model captures musical dependencies in time and pitch.

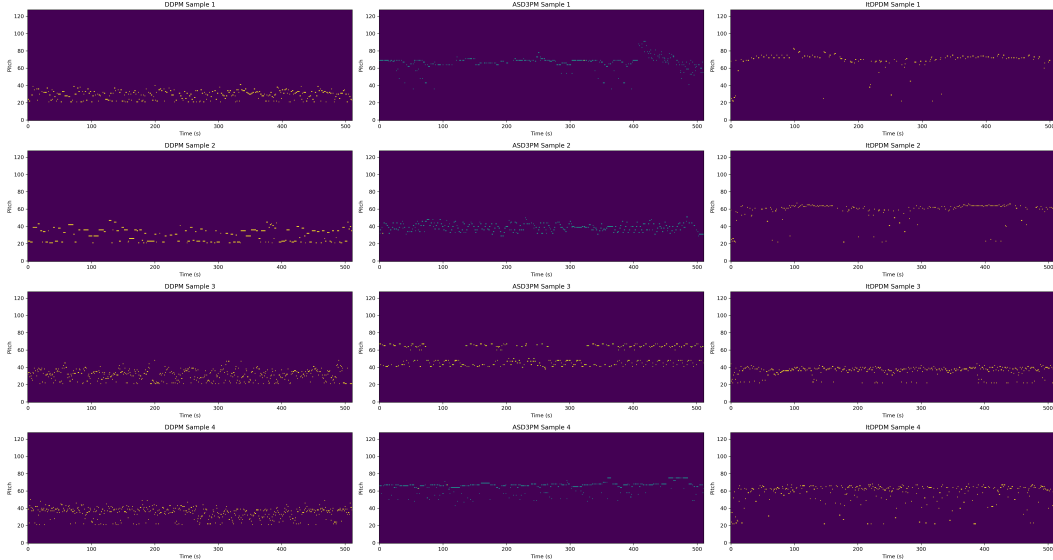


Figure 9: Piano roll visualizations of generated samples from DDPM (left), ASD3PM (middle), and ItDPDM (right). Each row corresponds to a particular random sample. Higher vertical positions represent higher pitches.

To further assess how the generated music matches the statistical properties of the training data, we also compare the generated pitch distributions with the ground truth. Figure 10 shows the histogram of MIDI pitch values for ItDPDM generated sequences alongside the empirical distribution from the training data with a close alignment indicating that the model captures global pitch statistics, such as register, range, and note density. Another observation is that in the generated samples, the note velocity is *slightly amplified* in comparison to the ground truth distribution.

C Synthetic Benchmark Details

C.1 Discrete benchmark details

We evaluate model performance on a suite of synthetic univariate discrete distributions designed to challenge generative models with features such as overdispersion, multimodality, sparsity, and skewness. All distributions take values in \mathbb{N}_0 and are non-negative.

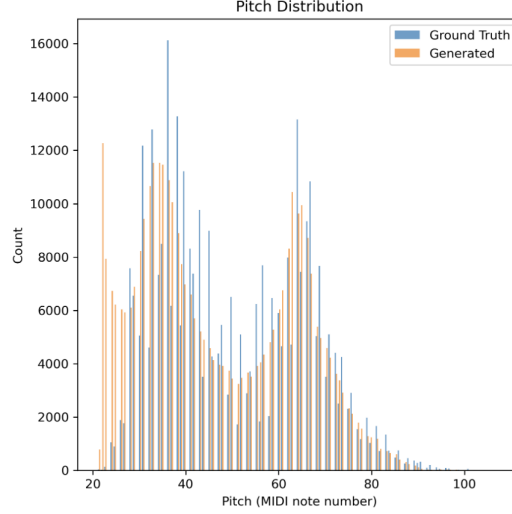


Figure 10: Comparing pitch distributions for ground truth and ItDPDM generated samples

1071 **Poisson Mixture (PoissMix):** This is a bimodal mixture of Poisson distributions:

$$0.1 \cdot \text{Poisson}(\lambda = 1) + 0.9 \cdot \text{Poisson}(\lambda = 100),$$

1072 producing a highly skewed and dispersed distribution with modes at both low and high counts,
1073 simulating tasks where most values are large but a minority remain near zero.

1074 **Zero-Inflated Poisson (ZIP):** To simulate data with an excess of zeros, we use a zero-inflated
1075 Poisson distribution: which samples zero with probability π_0 , and otherwise follows a Poisson
1076 distribution:

$$P(k) = \begin{cases} \pi_0 + (1 - \pi_0) \cdot e^{-\lambda}, & k = 0 \\ (1 - \pi_0) \cdot \frac{e^{-\lambda} \lambda^k}{k!}, & k > 0 \end{cases} \text{ with } \pi_0 = 0.7, \lambda = 5.$$

1077 This models structured sparsity common in count data with dropout.

1078 **Negative Binomial Mixture (NBinomMix):** This is a mixture of two negative binomial distribu-
1079 tions: $0.8 \cdot \text{NB}(1, 0.9) + 0.2 \cdot \text{NB}(10, 0.1)$, where the first mode has high probability near zero, while
1080 the second exhibits broader dispersion. It introduces skew and multimodality in count data.

1081 **Beta-Negative-Binomial (BNB):** The BNB distribution integrates a Beta prior over the success
1082 probability p of the negative binomial:

$$P(k) = \int_0^1 \text{NB}(k; 1, p) \cdot \text{Beta}(p; a = 1.5, b = 1.5) dp, \quad k \in \mathbb{N}_0.$$

1083 We use parameters $a = 0.5$, $b = 1.5$, and $r = 5$, inducing a heavy-tailed count distribution with
1084 long-range dependencies.

1085 **Zipf Distribution:** This power-law distribution is defined as:

$$P(x) = \frac{x^{-\alpha}}{\zeta(\alpha)}, \quad \alpha = 1.7$$

1086 , where $\zeta(\alpha)$ is the Riemann zeta function. Zipf distributions model naturally occurring frequencies,
1087 such as word counts or node degrees.

1088 **Yule–Simon Distribution:** The Yule–Simon distribution is defined as:

$$P(k) = \rho \cdot B(k, \rho + 1) = \rho \cdot \frac{\Gamma(k) \Gamma(\rho + 1)}{\Gamma(k + \rho + 1)}, \quad \rho = 2.0, k \in \mathbb{N}_{\geq 1},$$

where B is the Beta function and Γ is the gamma function. It is used to model data with power-law decay, often arising in preferential attachment or self-reinforcing (e.g. rich-get-richer) processes. These distributions form a challenging testbed for evaluating generative performance on discrete, non-negative data.

Table 5 summarizes the discrete synthetic benchmarks used in our study. Each distribution is selected to represent a different pathological regime—bi-modality, zero-inflation, overdispersion, or power-law behavior—intended to stress PMF concentration and test model robustness. For completeness, we specify parameter values used in generation and annotate tail behaviors to clarify their impact on sample complexity and generalization.

Distribution	Parameters	Tail behaviour
PoissMix	$\lambda = \{1, 100\}$	bi-modal
Zero-Inflated Poisson	$\pi_0 = 0.7, \lambda = 5$	spike at 0
NBinomMix	$(r, p) = \{(1, 0.9), (10, 0.1)\}$	$\text{Var} > \text{E}$
BNB	$a = 0.5, b = 1.5, r = 5$	power-law
Zipf	$\alpha = 1.7$	$\sim x^{-\alpha}$
Yule-Simon	$\rho = 2.0$	heavier than Zipf

Table 5: Specification of discrete synthetic benchmarks. All distributions are heavy-tailed, zero-inflated, or multi-modal, stressing PMF concentration.

C.2 Training Details & Metrics

In addition to the ConditionalMLP, a timestep embedding network additionally projects diffusion steps into a 64-dimensional space using SiLU activations. Models are trained for 200 epochs using the Adam optimizer ($\eta = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with a batch size of 128. The Gaussian DDPM employs a linear noise schedule $\beta_t \in [10^{-4}, 2 \cdot 10^{-2}]$ over $T = 100$ diffusion steps. Our ItDPDM framework adopts a linear gamma schedule $\gamma_t \in [1.0, 0.0]$ over the same number of steps. For Poisson diffusion, the initial sample mean is set to 10.0.

Wasserstein-1 distance Wasserstein-1 distance [54] between two univariate distributions p and q is defined as: $W_1(p, q) := \int_{\mathbb{R} \times \mathbb{R}} |x - y| d\pi(x, y) = \int_{\mathbb{R}} |P(x) - Q(x)| dx$, where $\pi(x, y)$ is a joint coupling of p and q , and P, Q are their respective cumulative distribution functions (CDFs). When p and q are empirical distributions of the same size n , this reduces to: $W_1(p, q) = \frac{1}{n} \|\text{sort}(X) - \text{sort}(Y)\|_1$, where $X, Y \in \mathbb{R}^n$ are the sorted samples from p and q .

For each empirical distribution of 50,000 generated samples over 5 runs, say \hat{p}_{gen} (with \hat{p}_{test} denoting the empirical distribution of 50,000 test samples), we compute the Wasserstein-1 distance (WD) [54] and negative log-likelihood (NLL) as:

$$\text{WD} = W_1(\hat{p}_{\text{test}}, \hat{p}_{\text{gen}}), \text{NLL} = -\frac{1}{n_{\text{test}}} \sum_i \log \hat{p}_{\text{gen}}(x_i) \quad (24)$$

where x_i denote the held-out samples.

C.3 Probability Mass Function Estimation:

For discrete distributions, we estimate the empirical probability mass function (PMF) $\hat{p}(x)$ from generated samples $\{x_i\}_{i=1}^N$ using a histogram-based approach with binning over a finite support $\mathcal{X} = \{0, 1, \dots, K\}$:

$$\hat{p}(x) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(x_i = x), \quad (25)$$

where $\mathbb{I}(\cdot)$ is the indicator function and K is the truncation value. We set $K = 50$ across all experiments to standardize the support. To reduce sampling noise and better visualize differences across models, we additionally compute a *smoothed PMF estimate* using a discrete Gaussian kernel:

$$\hat{p}_{\text{smooth}}(x) = \frac{1}{N} \sum_{i=1}^N K_h(x - x_i), \quad (26)$$

1121 where $K_h(\cdot)$ is a Gaussian kernel defined on the integer lattice:

$$K_h(x) = \frac{1}{Z} \exp\left(-\frac{x^2}{2h^2}\right), \quad (27)$$

1122 with normalization constant $Z = \sum_{x' \in \mathcal{X}} \exp\left(-\frac{x'^2}{2h^2}\right)$ ensuring that K_h sums to 1 over the support.
 1123 The bandwidth h is selected empirically per distribution to balance smoothness and fidelity to the
 1124 empirical histogram. To assess variability in PMF estimation, we also compute error bands via
 1125 non-parametric bootstrapping. Specifically, we generate 10 bootstrap resamples of the model outputs,
 1126 re-estimate the (smoothed) PMF for each, and plot the mean \pm standard deviation across these
 1127 resampled estimates. Each plot includes in Fig. 6 includes: a) ground-truth PMF (when known), and
 1128 b) the empirical unsmoothed and smoothed PMFs for each model (e.g., ItDPDM, DDPM, LTJ), with
 1129 any shaded error bands reflecting bootstrap variability.

Implementation Details:

Aspect	Details
Sample size	$N = 10,000$ samples per model and distribution
Support	$\mathcal{X} = \{0, 1, \dots, 50\}$ for discrete; bounded x for continuous
Smoothing bandwidth	h tuned per distribution (discrete); KDE bandwidth default
Bootstrap	10 resamples per model for uncertainty estimation
Visualization	True distribution, model estimates, and error bands plotted

Table 6: Summary of implementation settings for PMF and PDF estimation.

1130

1131 **Zoomed-in look at PMF plots:** Building on the analysis in Section 5, Figure 11 and Figure 12
 1132 provides a magnified view of the Yule–Simon and Zipf fits produced by each model. ItDPDM exhibits
 1133 the closest alignment to the target distribution, particularly in the critical low-support region.

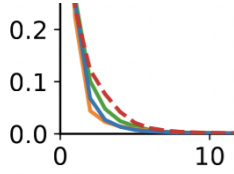


Figure 11: Zoomed-in Yule–Simon fits

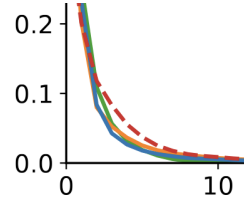


Figure 12: Zoomed-in Zipf law fits

1134 C.4 Non-negative Continuous Scenarios

1135 As stated earlier, we extend our analysis to six skewed continuous densities: Gamma, Log-Normal,
 1136 Lomax, Half-Cauchy, Half-t, Weibull, (along with Beta and Uniform distributions) as outlined in this
 1137 section. Our goal here is to assess how well generative models capture asymmetry, concentration,
 1138 and long-range dependencies in continuous data.

1139 Descriptions and parameters:

1140 **Gamma Distribution:** The Gamma distribution is defined by a shape parameter ‘ a ’ and a scale
 1141 parameter ‘ θ ’:

$$p(x) = \frac{1}{\Gamma(a)\theta^a} x^{a-1} e^{-x/\theta}, \quad x \geq 0.$$

1142 We use $a = 0.5$, $\theta = 2$, which produces a sharp mode near zero and a long right tail. Gamma
 1143 distributions are commonly used to model wait times, energy release, and insurance claims—making
 1144 them valuable for stress-testing the model’s handling of high variance and positive skew.

1145 **Log-Normal Distribution:** A log-normal distribution arises when the logarithm of a variable is
 1146 normally distributed:

$$p(x) = \frac{1}{xs\sqrt{2\pi}} \exp\left(-\frac{(\log x - \mu)^2}{2s^2}\right), \quad x > 0.$$

1147 We use $\mu = 0, s = 1.5$, producing a distribution with significant positive skew and heavy tails. Log-
 1148 normal models appear in financial returns, biological measurements, and natural language modeling,
 1149 where multiplicative effects dominate.

1150 **Lomax Distribution:** Also known as the Pareto Type II distribution, the Lomax is defined as:

$$p(x) = \frac{c}{s} \left(1 + \frac{x}{s}\right)^{-(c+1)}, \quad x \geq 0.$$

1151 We use $c = 2.0, s = 1.0$, resulting in a fat-tailed distribution often used in reliability engineering and
 1152 modeling rare, catastrophic events. It challenges models to capture high-probability mass near zero
 1153 with occasional large outliers.

1154 **Half-Cauchy Distribution:** The Half-Cauchy is the positive part of a Cauchy distribution:

$$p(x) = \frac{2}{\pi s \left[1 + \left(\frac{x}{s}\right)^2\right]}, \quad x \geq 0.$$

1155 With $s = 1$, this distribution has undefined mean and variance, and extremely heavy tails. It is
 1156 commonly used as a prior in hierarchical Bayesian models due to its robustness to outliers.

1157 **Half-t Distribution:** The Half- t distribution is the absolute value of a Student's t -distributed variable:

$$p(x) = 2 \cdot t(x; \nu, 0, s), \quad x \geq 0.$$

1158 We use $\nu = 3, s = 1$, yielding a distribution with heavy but finite tails. This is another robust
 1159 prior used in Bayesian inference, particularly for variances in hierarchical models, where it prevents
 1160 over-shrinkage.

1161 **Weibull Distribution:** The Weibull distribution, defined by shape k and scale λ , is given by:

$$p(x) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, \quad x \geq 0.$$

1162 We use $k = 1.5, \lambda = 1$, producing a distribution with increasing hazard rate and moderate skew. This
 1163 is widely used in survival analysis, material failure modeling, and wind speed distributions.

1164 **Beta Distribution (bounded support):** Though often used on $[0, 1]$, the Beta distribution provides
 1165 diverse shapes depending on the parameters:

$$p(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}, \quad 0 \leq x \leq 1.$$

1166 We use $a = 2, b = 2$, leading to a density concentrated near zero. The Beta distribution tests
 1167 the model's ability to learn bounded distributions with asymmetric mass concentration, relevant in
 1168 probabilistic modeling and reinforcement learning. A key limitation to note here is that in case of
 1169 asymmetric/skewed beta distributions, all the models notably fail to learn the distribution.

1170 **Uniform Distribution (flat support):** The uniform distribution provides a baseline for bounded,
 1171 structureless densities:

$$p(x) = \frac{1}{b-a}, \quad a \leq x \leq b.$$

1172 We set $a = 0, b = 1$, resulting in a constant density over the unit interval. Although simple, it serves
 1173 as a sanity check for model calibration and ability to avoid mode collapse under flat distributions.
 1174 Together, these distributions offer a comprehensive testbed for evaluating generative modeling under
 1175 varied support, skewness, and tail behavior. They also represent common scenarios encountered in
 1176 practice, ensuring relevance to real-world generative tasks.

1177 Results:

1178 Table 7 compares the Wasserstein distance for all the continuous cases, and in the continuous case,
 1179 we omit NLL values as they can be overly sensitive to skewness and outliers, making them unreliable
 1180 for fair comparison. More critically, whereas the true NLL in continuous distributions can often be
 1181 negative while our discrete estimator cannot possibly yield a negative NLL.

1182 For each distribution, we visualize the estimated PDFs from all models alongside the true density.
 1183 Figure 13 summarizes the results across all eight distributions, providing a qualitative comparison of
 1184 how closely each model recovers the underlying data-generating process.

Table 7: WD for continuous cases (\downarrow lower is better). Bold indicates best.

Distribution	WD		
	DDPM	hDDPM	LTJ
Gamma	0.27 ± 0.09	0.12 ± 0.05	0.14 ± 0.05
Log-Normal	2.39 ± 0.53	1.94 ± 0.71	1.99 ± 0.66
Lomax	0.39 ± 0.20	0.31 ± 0.17	1.15 ± 0.41
Half-Cauchy	6.67 ± 2.45	6.35 ± 2.56	5.45 ± 2.23
Half-t	0.20 ± 0.07	0.21 ± 0.02	0.22 ± 0.04
Weibull	0.29 ± 0.05	0.23 ± 0.02	0.23 ± 0.06
Beta	0.28 ± 0.07	0.18 ± 0.03	0.19 ± 0.06
Uniform	0.12 ± 0.05	0.12 ± 0.03	0.12 ± 0.02

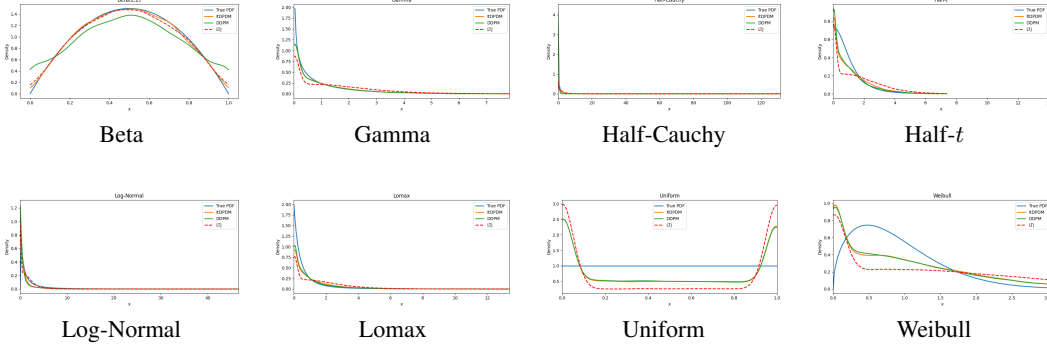


Figure 13: Comparison of estimated PDFs for various continuous distributions in the synthetic dataset. Each plot shows the true distribution and model-generated estimates.

1185 PDF estimation:

1186 For continuous non-negative distributions, we estimate the probability density function (PDF) $\hat{f}(x)$
 1187 using kernel density estimation (KDE) with a Gaussian kernel:

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{(x-x_i)^2}{2h^2}\right), \text{ where by default } h = \sigma N^{-1/(d+4)}, d=1 \Rightarrow h = \sigma N^{-1/5}, \quad (28)$$

1188 with σ denoting the sample standard deviation of $\{x_i\}$ and N the number of samples.

1189 We compute error bands by bootstrapping: for each model, we resample its generated samples
 1190 10 times, compute the KDE for each resample, and display the mean \pm standard deviation across
 1191 estimates. For bounded distributions (e.g., Beta, Uniform), we clip model-generated samples to the
 1192 distribution’s support before applying KDE. Each PDF plot includes: a) ground-truth PDF, and b) the
 1193 average KDE for each model, with any shaded error bands indicating bootstrap uncertainty.

1194 D Section 3 Proofs

1195 D.1 On the Poisson Loss Function:

1196 Here, as outlined in 3.2, we establish that the function $l_0(x) = x \log x - x + 1$ serves as the convex
 1197 conjugate of the Poisson distribution’s log moment generating function (log MGF). We begin by
 1198 deriving the log MGF of the Poisson distribution, and finally computing its convex conjugate through
 1199 the Legendre-Fenchel transform. Let X be a random variable following a Poisson distribution with
 1200 parameter $\lambda > 0$. The probability mass function (PMF) of X is given by:

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad \text{for } k = 0, 1, 2, \dots$$

1201 The moment generating function (MGF) can be evaluated as:

$$M_X(t) = E[e^{tX}] = \sum_{k=0}^{\infty} e^{tk} P(X = k) = \sum_{k=0}^{\infty} e^{tk} \frac{\lambda^k e^{-\lambda}}{k!} = e^{-\lambda} \sum_{k=0}^{\infty} \frac{(\lambda e^t)^k}{k!} = e^{-\lambda} e^{\lambda e^t} = e^{\lambda(e^t - 1)}$$

1202 Let $\phi(t)$ be the log moment generating function as shown:

$$\phi(t) = \log M_X(t) = \lambda(e^t - 1)$$

1203 Without any loss of generality, let $\lambda = 1$ (since scaling does not affect the form of the conjugate),
 1204 implying $\phi(t) = e^t - 1$. The **convex conjugate** of a convex function $\phi : \mathbb{R} \rightarrow \mathbb{R} \cup \{+\infty\}$, denoted
 1205 by $\phi^*(x)$, is defined as:

$$\phi^*(x) = \sup_{t \in \mathbb{R}} \{xt - \phi(t)\}$$

1206 This transformation maps the original function $\phi(t)$ to its dual function $\phi^*(x)$, and then finds the
 1207 supremum of linear functions subtracted by $\phi(t)$.

1208 Let $\phi(t) = e^t - 1$ be the log moment generating function (log MGF) of a Poisson distribution with
 1209 parameter $\lambda = 1$. Then, the convex conjugate of ϕ , denoted by $\phi^*(x)$, is given by:

$$\phi^*(x) = \begin{cases} x \log x - x + 1 & \text{if } x > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

1210 *Proof.* By definition: $\phi^*(x) = \sup_{t \in \mathbb{R}} \{xt - \phi(t)\} = \sup_{t \in \mathbb{R}} \{xt - e^t + 1\}$

1211 To find the supremum, we find the value of t that maximizes this expression. First-order conditions
 1212 imply: $\frac{d}{dt}(xt - e^t) = x - e^t = 0$ so we have $t = \log x$. This critical point exists only if $x > 0$, as
 1213 $e^t > 0$ for all $t \in \mathbb{R}$. From the second-order condition, we get:

$$\frac{d^2}{dt^2}(xt - e^t) = -e^t < 0 \quad \forall t \in \mathbb{R}$$

1214 The negative second derivative confirms that the function is concave at $t = \log x$, ensuring a global
 1215 maximum at this point. So for $t = \log x$,

$$\phi^*(x) = x(\log x) - e^{\log x} + 1 = x \log x - x + 1$$

1216 Therefore, for $x > 0$:

$$\phi^*(x) = x \log x - x + 1$$

1217 For $x \leq 0$, the supremum is unbounded above, leading to: $\phi^*(x) = +\infty$ Combining these cases
 1218 gives:

$$\phi^*(x) = \begin{cases} x \log x - x + 1 & \text{if } x > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

1219 This establishes that $l_0(x) = x \log x - x + 1$ is the convex conjugate of the Poisson distribution's
 1220 log moment generating function $\phi(t) = e^t - 1$ and therefore, a natural loss function.

1221 Connection to Bregman Divergence

1222 The Poisson loss function we defined $l(x, \hat{x})$ is a member of the broader family of Bregman diver-
 1223 gences, which are pivotal in various domains such as machine learning, information theory, and
 1224 optimization. A Bregman divergence is defined for a strictly convex and differentiable function
 1225 $\psi : \mathbb{R}^d \rightarrow \mathbb{R}$ as follows:

$$\mathcal{L}_\psi(x, \hat{x}) = \psi(x) - \psi(\hat{x}) - \langle \nabla \psi(\hat{x}), x - \hat{x} \rangle,$$

1226 where $\langle \cdot, \cdot \rangle$ denotes the inner product in \mathbb{R}^d , and $\nabla \psi(\hat{x})$ represents the gradient of ψ evaluated at \hat{x} .

1227 For the Poisson loss function, the generating function ψ is chosen as:

$$\psi(x) = x \log x - x.$$

1228 Substituting this into the Bregman divergence definition yields:

$$\mathcal{L}_\psi(x, \hat{x}) = x \log x - x - (\hat{x} \log \hat{x} - \hat{x}) - (\log \hat{x} \cdot (x - \hat{x})).$$

1229 Simplifying the expression, we obtain:

$$\mathcal{L}_\psi(x, \hat{x}) = x \log\left(\frac{x}{\hat{x}}\right) - x + \hat{x},$$

1230 which is precisely the Poisson loss function $l(x, \hat{x})$.

1231 This framework not only encapsulates the Poisson loss but also generalizes it to encompass other
 1232 widely-used loss functions by merely altering the generating function ψ . Well-known examples
 1233 include squared error loss (choosing $\psi(x) = \frac{1}{2}x^2$ and Itakura-Saito divergence (choosing $\psi(x) =$
 1234 $-\log x$). Bregman divergences exhibit key properties that make them valuable in optimization and
 1235 learning. They are **non-negative**, vanishing only when $x = \hat{x}$, due to the strict convexity of ψ .
 1236 They are also **asymmetric**, meaning $\mathcal{L}_\psi(x, \hat{x}) \neq \mathcal{L}_\psi(\hat{x}, x)$ in general and their **projection property**
 1237 enables efficient optimization over convex sets.

1238 By leveraging the Bregman divergence framework, Poisson and Gaussian diffusion schemes can be
 1239 unified under a single theoretical umbrella, where squared error loss ($\psi(x) = \frac{1}{2}x^2$) corresponds to
 1240 Gaussian noise, and Poisson loss aligns with count-based data modeling. This unification enables
 1241 extending optimization techniques across different noise models by adjusting the generating function
 1242 ψ . Viewing Poisson loss function as a Bregman divergence thus broadens its theoretical and practical
 1243 utility discrete data modelling.

1244 Optimality of Conditional Expectation

1245 Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ be a strictly convex and differentiable function. The Bregman divergence D_ϕ
 1246 induced by ϕ is defined by

$$D_\phi(X, Y) = \phi(X) - \phi(Y) - \nabla\phi(Y)^\top(X - Y).$$

1247 Consider a random variable $X \in \mathbb{R}^d$ and a sigma-algebra $\sigma(Z)$ with $Y = Y(Z)$ being any measurable
 1248 function of Z . Let $Y^* = E[X|Z]$ denote the conditional expectation of X given Z . The objective is
 1249 to show that Y^* uniquely minimizes the expected Bregman loss $E[D_\phi(X, Y)]$ among all measurable
 1250 functions $Y(Z)$. For any such function Y , consider the difference in expected Bregman losses:

$$E[D_\phi(X, Y)] - E[D_\phi(X, Y^*)] = E[\phi(X) - \phi(Y) - \nabla\phi(Y)^\top(X - Y)] - E[\phi(X) - \phi(Y^*) - \nabla\phi(Y^*)^\top(X - Y^*)]$$

1251 Simplifying, the terms involving $\phi(X)$ cancel out, yielding

$$E[D_\phi(X, Y)] - E[D_\phi(X, Y^*)] = E[\phi(Y^*) - \phi(Y) - \nabla\phi(Y)^\top(Y^* - Y)].$$

1252 Recognizing that Y^* is the conditional expectation $E[X|Z]$, we utilize the law of total expectation to
 1253 express the above as

$$E[\phi(Y^*) - \phi(Y) - \nabla\phi(Y)^\top(Y^* - Y)] = E[D_\phi(Y^*, Y)].$$

1254 Due to the strict convexity of ϕ , the Bregman divergence satisfies $D_\phi(u, v) \geq 0$ for all $u, v \in \mathbb{R}^d$,
 1255 with equality if and only if $u = v$. Therefore,

$$E[D_\phi(X, Y)] - E[D_\phi(X, Y^*)] = E[D_\phi(Y^*, Y)] \geq 0,$$

1256 with equality holding if and only if $Y = Y^*$ almost surely. This establishes that

$$E[D_\phi(X, Y)] \geq E[D_\phi(X, Y^*)],$$

1257 for all measurable functions $Y(Z)$, and thus $Y^* = E[X|Z]$ is the unique minimizer of the expected
 1258 Bregman loss $E[D_\phi(X, Y)]$.

1259 D.2 Section 3 Lemma Proofs

1260 **Proof of Lemma 1: Properties of Poisson Loss** Consider the loss function defined as $l(x, \hat{x}) =$
 1261 $\hat{x} \cdot l_0\left(\frac{x}{\hat{x}}\right)$, where $l_0(z) = z \log z - z + 1$.

1262 **1. Non-negativity:** Since $l_0(z)$ achieves its minimum value of 0 at $z = 1$ and is non-negative for all
 1263 $z > 0$, it follows that $l(x, \hat{x}) \geq 0$ for all $x, \hat{x} > 0$. Equality holds if and only if $\frac{x}{\hat{x}} = 1$, i.e., $x = \hat{x}$.

1264 **2. Convexity:** The function $l_0(z)$ is convex in z because its second derivative $l_0''(z) = \frac{1}{z}$ is positive
 1265 for all $z > 0$. Therefore, $l(x, \hat{x}) = \hat{x} \cdot l_0\left(\frac{x}{\hat{x}}\right)$ is convex in \hat{x} for each fixed x , and similarly, it is convex
 1266 in x for each fixed \hat{x} , as the composition of a convex function with an affine transformation preserves
 1267 convexity. (We can also directly use the Bregman divergence framework to argue its convexity)

1268 **3. Scaling:** For any $\alpha > 0$, consider scaling both arguments of the loss function:

$$l(\alpha x, \alpha \hat{x}) = \alpha \hat{x} \cdot l_0\left(\frac{\alpha x}{\alpha \hat{x}}\right) = \alpha \hat{x} \cdot l_0\left(\frac{x}{\hat{x}}\right) = \alpha \cdot l(x, \hat{x}).$$

1269 This demonstrates that the loss function scales linearly with α .

1270 **4. Unboundedness for Underestimation:** For any fixed $x > 0$, as $\hat{x} \rightarrow 0^+$, the ratio $\frac{x}{\hat{x}} \rightarrow \infty$.
 1271 Evaluating the loss function in this limit:

$$l(x, \hat{x}) = \hat{x} \cdot \left(\frac{x}{\hat{x}} \log\left(\frac{x}{\hat{x}}\right) - \frac{x}{\hat{x}} + 1 \right) = x \log\left(\frac{x}{\hat{x}}\right) - x + \hat{x}.$$

1272 As $\hat{x} \rightarrow 0^+$, $\log\left(\frac{x}{\hat{x}}\right)$ grows without bound, causing $l(x, \hat{x}) \rightarrow \infty$. This shows that the loss becomes
 1273 unbounded as \hat{x} underestimates x .

1274 **Proof of Lemma 2.** Let Z_γ be a Poisson random variable with parameter γX , meaning $Z_\gamma|X =$
 1275 $x \sim \text{Pois}(\gamma x)$. Suppose the conditional expectation $\langle X \rangle_z = E[X|Z_\gamma = z]$ is affine in z ,

$$\langle X \rangle_z = az + b,$$

1276 for some a and b , with $0 < a < 1/\gamma$ and $b > 0$. We aim to show that X follows a Gamma distribution
 1277 with shape $\alpha = \frac{1-\gamma a}{a}$ and rate $\beta = \frac{a}{b}$, i.e.,

$$X \sim \text{Gamma}\left(\frac{1-\gamma a}{a}, \frac{a}{b}\right).$$

1278 Define $U = X$ and $Y = Z_\gamma \sim \mathcal{P}(\gamma U)$. Assume $E[U|Y = z] = az + b$. By the law of total
 1279 expectation,

$$0 = E[(U - (aY + b))g(Y)]$$

1280 for any function g satisfying integrability. Choosing $g(Y) = e^{-tY}$ for $t > 0$,

$$E[(U - (aY + b))e^{-tY}] = 0.$$

1281 Rewriting $Y \sim \mathcal{P}(\gamma U)$, we use the known conditional Laplace transform relation for a $\mathcal{P}(\lambda)$ random
 1282 variable Y ,

$$E[e^{-tY}|U = u] = \exp(u(\gamma(e^{-t}) - 1)).$$

1283 Hence,

$$E[e^{-tY}] = E[\exp(U\gamma(e^{-t}) - 1)],$$

1284 which is the Laplace transform of U evaluated at $s = \gamma(1 - e^{-t})$. Denote

$$L_U(s) = E[e^{-sU}], \quad \text{so that} \quad E[e^{-tY}] = L_U(\gamma(1 - e^{-t})).$$

1285 Similarly,

$$E[Ue^{-tY}] = -\frac{d}{ds}L_U(s)\Big|_{s=\gamma(1-e^{-t})}, \quad E[Ye^{-tY}] = -\frac{d}{dt}E[e^{-tY}].$$

1286 From the orthogonality condition,

$$E[(U - (aY + b))e^{-tY}] = 0.$$

1287 Using the above expressions,

$$0 = E[Ue^{-tY}] - aE[Ye^{-tY}] - bE[e^{-tY}].$$

1288 Substituting $s = \gamma(1 - e^{-t})$ and differentiating as needed, we obtain a first-order linear differential
 1289 equation for $L_U(s)$,

$$-((1 - a\gamma) + a\gamma s)L_U'(s) = bL_U(s).$$

1290 The unique solution with $L_U(0) = 1$ is

$$L_U(s) = \left(1 + \frac{b}{1 - \gamma a} s\right)^{-\frac{1-\gamma a}{a}}.$$

1291 This is the Laplace transform of a $\text{Gamma}\left(\frac{1-\gamma a}{a}, \frac{a}{b}\right)$ random variable. Hence, $U = X$ follows this
 1292 Gamma distribution. For the Gamma distribution to be well-defined with a positive shape parameter,
 1293 we require $\alpha = \frac{1-\gamma a}{a} > 0$, which holds for $0 < a < \frac{1}{\gamma}$. The rate parameter $\beta = \frac{a}{b} > 0$ requires
 1294 $b > 0$. Under these conditions, $X \sim \text{Gam}\left(\frac{1-\gamma a}{a}, \frac{a}{b}\right)$, completing the proof.

1295 **Proof of Lemma 3.** When $X = 0$ almost surely, $E[X] = 0$, and the identity holds by convention.
 1296 Else, $E[X] > 0$, and we have:

$$\begin{aligned} E[l(X, \hat{x})] &= E\left[X \log\left(\frac{X}{\hat{x}}\right) - X + \hat{x}\right] = E[X \log X] - E[X \log \hat{x}] - E[X] + \hat{x} \\ &= E[X \log X - X \log E[X] - X + E[X]] + l(E[X], \hat{x}) \\ &= E[l(X, E[X])] + l(E[X], \hat{x}). \end{aligned}$$

1297 **Proof of Lemma 4.** Consider $Z_\gamma = \mathcal{P}(\gamma X)$, where Z_γ is a Poisson random variable with parameter
 1298 γX . To determine $\langle X \rangle_z = E[X|Z_\gamma = z]$ for each $z \geq 0$, we start by applying the definition of
 1299 conditional expectation:

$$\langle X \rangle_z = \frac{E[X \cdot P_{Z_\gamma}(z|X)]}{P_{Z_\gamma}(z)}.$$

1300 Given that $Z_\gamma|X = x \sim \text{Pois}(\gamma x)$, the conditional probability mass function is

$$P_{Z_\gamma}(z|X = x) = \frac{(\gamma x)^z e^{-\gamma x}}{z!}.$$

1301 Substituting this into the expression for $\langle X \rangle_z$ yields

$$\langle X \rangle_z = \frac{E\left[X \cdot \frac{(\gamma X)^z e^{-\gamma X}}{z!}\right]}{P_{Z_\gamma}(z)}.$$

1302 To relate $\langle X \rangle_z$ to $P_{Z_\gamma}(z+1)$, observe that

$$P_{Z_\gamma}(z+1) = E\left[\frac{(\gamma X)^{z+1} e^{-\gamma X}}{(z+1)!}\right] = \frac{\gamma}{z+1} E\left[X \cdot \frac{(\gamma X)^z e^{-\gamma X}}{z!}\right].$$

1303 Rearranging the above equation, we obtain

$$E\left[X \cdot \frac{(\gamma X)^z e^{-\gamma X}}{z!}\right] = \frac{(z+1)}{\gamma} P_{Z_\gamma}(z+1).$$

1304 Substituting this back into the expression for $\langle X \rangle_z$, we have

$$\langle X \rangle_z = \frac{\frac{(z+1)}{\gamma} P_{Z_\gamma}(z+1)}{P_{Z_\gamma}(z)} = \frac{1}{\gamma} \frac{(z+1) P_{Z_\gamma}(z+1)}{P_{Z_\gamma}(z)}.$$

1305 This completes the proof of Lemma 4.

1306 The conditional expectation over a Poisson noise channel also has other unique properties, some of
 1307 which are stated below. The next property is useful in showing that the conditional expectation in this
 1308 case is unique for every input distribution.

1309 **Lemma 5.** Let $Z_\gamma = \mathcal{P}(\gamma X)$. Then, for every positive integer k and every non-negative integer z ,

$$E[(\gamma X)^k | Z_\gamma = z] = \prod_{i=0}^{k-1} E[\gamma X | Z_\gamma = z + i].$$

1310 **Proof of Lemma 5.** Let $Z_\gamma = \mathcal{P}(\gamma X)$. We claim that for every positive integer k and nonnegative
 1311 integer z ,

$$E[(\gamma X)^k | Z_\gamma = z] = \prod_{i=0}^{k-1} E[\gamma X | Z_\gamma = z + i].$$

1312 From the affine formula in Lemma 4, the conditional expectation of γX given $Z_\gamma = z$ is related to
 1313 the ratio of marginal probabilities. More generally, for higher-order moments,

$$E[(\gamma X)^k | Z_\gamma = z] = \frac{(z+k)!}{z!} \frac{P_{Z_\gamma}(z+k)}{P_{Z_\gamma}(z)}. \quad (29)$$

1314 We can also express $(\gamma X)^k$ as a product of γX terms and use the Poisson shifting property of $\mathcal{P}(\gamma X)$.
 1315 Applying Lemma 4 and Eq. 29 for each shift $z \rightarrow z + i$ gives

$$E[(\gamma X)^k | Z_\gamma = z] = \prod_{i=0}^{k-1} E[\gamma X | Z_\gamma = z + i].$$

1316 Each factor on the right captures the conditional expectation of γX at consecutive levels $z, z +$
 1317 $1, \dots, z + k - 1$, so all higher-order moments of γX follow from the first conditional moment
 1318 $E[\gamma X | Z_\gamma = z]$. This completes the proof.

1319 *Proof Sketch of Eq. 29:* The key observation behind the formula is that, for the Poisson distribution,
 1320 shifting from y to $y + k$ multiplies the corresponding probability mass by $\frac{(aX + \lambda)^k}{k!}$. Evaluating
 1321 the expectation leverages the ratio of adjacent Poisson probabilities $P_Y(y + k)/P_Y(y)$ and tracks
 1322 how $(aX + \lambda)^k$ factors. In essence, a product expansion shows how each additional factor $aX + \lambda$
 1323 increases the count from y to $y + 1$, and iterating this argument recovers the moment expression. As
 1324 shown in [57], for Poisson observations $Z_\gamma \sim \mathcal{P}(aX + \lambda)$, the sequence of conditional expectations
 1325 $\{\mathbb{E}[X | Z_\gamma = z]\}_{z \geq 0}$ uniquely determines the input distribution P_X . This supports our information-
 1326 theoretic derivation and strengthens the foundation for learning in discrete-state noise models. For
 1327 our Poisson setting, we also have:

1328 **Lemma 6.** Let $Z_\gamma = \mathcal{P}(\gamma X)$. Then, for every $\gamma > 0$ and $y = 0, 1, \dots$,

$$\frac{d}{d\gamma} P_{Z_\gamma|X}(y|x) = x (P_{Z_\gamma|X}(y-1|x) - P_{Z_\gamma|X}(y|x)), \quad \gamma \frac{d}{d\gamma} P_{Z_\gamma}(y) = y P_{Z_\gamma}(y) - (y+1) P_{Z_\gamma}(y+1)$$

1329 where $P_{Z_\gamma|X}(-1|x) = P_{Z_\gamma}(-1) = 0$.

1330 **Proof of Lemma 6.** Let $Z_\gamma = \mathcal{P}(\gamma X)$, where Z_γ is a Poisson random variable with parameter γX .
 1331 We first compute the derivative of the conditional probability mass function $P_{Z_\gamma}(z|X = x)$ with
 1332 respect to γ .

1333 Since Z_γ given $X = x$ follows a Poisson distribution with mean γx , we have

$$P_{Z_\gamma}(z|X = x) = \frac{(\gamma x)^z e^{-\gamma x}}{z!}.$$

1334 Taking the derivative with respect to γ and using product rule, we obtain:

$$\frac{d}{d\gamma} P_{Z_\gamma}(z|X = x) = \frac{d}{d\gamma} \left(\frac{(\gamma x)^z e^{-\gamma x}}{z!} \right) = \frac{z(\gamma x)^{z-1} x e^{-\gamma x}}{z!} - \frac{x(\gamma x)^z e^{-\gamma x}}{z!}.$$

1335 Simplifying the terms, we obtain

$$\frac{d}{d\gamma} P_{Z_\gamma}(z|X = x) = x \left(\frac{(\gamma x)^{z-1} e^{-\gamma x}}{(z-1)!} - \frac{(\gamma x)^z e^{-\gamma x}}{z!} \right).$$

1336 Notice that

$$\frac{(\gamma x)^{z-1} e^{-\gamma x}}{(z-1)!} = P_{Z_\gamma}(z-1|X = x),$$

1337 we can rewrite the derivative as

$$\frac{d}{d\gamma} P_{Z_\gamma}(z|X = x) = x (P_{Z_\gamma}(z-1|X = x) - P_{Z_\gamma}(z|X = x)).$$

1338 This establishes the first part of the lemma.

1339 Next, we compute the derivative of the marginal probability $P_{Z_\gamma}(z)$ with respect to γ . By the law of
 1340 total probability, we have

$$P_{Z_\gamma}(y) = E [P_{Z_\gamma}(z|X)].$$

1341 Differentiating both sides with respect to γ , we obtain

$$\frac{d}{d\gamma} P_{Z_\gamma}(z) = E \left[\frac{d}{d\gamma} P_{Z_\gamma}(z|X) \right].$$

1342 Substituting the result from above, we get

$$\frac{d}{d\gamma} P_{Z_\gamma}(z) = E \left[x \left(P_{Z_\gamma}(z-1|X) - P_{Z_\gamma}(z|X) \right) \right].$$

1343 This can be expressed as

$$\gamma \frac{d}{d\gamma} P_{Z_\gamma}(z) = \gamma E \left[x P_{Z_\gamma}(z-1|X) \right] - \gamma E \left[x P_{Z_\gamma}(z|X) \right].$$

1344 Noting that for a Poisson distribution, $E \left[x P_{Z_\gamma}(z|X) \right] = \frac{z}{\gamma} P_{Z_\gamma}(z)$ and $E \left[x P_{Z_\gamma}(z-1|X) \right] =$
 1345 $\frac{z}{\gamma} P_{Z_\gamma}(z)$, we substitute to obtain

$$\gamma \frac{d}{d\gamma} P_{Z_\gamma}(z) = z P_{Z_\gamma}(z) - (z+1) P_{Z_\gamma}(z+1).$$

1346 Thus, the second part of the lemma is established.

1347 Other properties of the Conditional Expectation

1348 **Lemma 7.** Let $Z_\gamma = \mathcal{P}(\gamma X)$ where X is a nonnegative random variable, and $\gamma > 0$. Then, for
 1349 every $\gamma > 0$ and integer $z \geq 0$,

$$\frac{d}{d\gamma} E[X|Z_\gamma = z] = -z\gamma \text{Var}(X|Z_\gamma = z-1),$$

1350 where $\text{Var}(X|Z_\gamma = -1) = 0$.

1351 *Proof.* Fix an integer $z \geq 0$. Consider the conditional expectation

$$E[X|Z_\gamma = z] = \frac{1}{\gamma} \left((z+1) \frac{P(Z_\gamma = z+1)}{P(Z_\gamma = z)} \right).$$

1352 Differentiating both sides with respect to γ , we obtain

$$\frac{d}{d\gamma} E[X|Z_\gamma = z] = \frac{1}{\gamma} \frac{d}{d\gamma} \left((z+1) \frac{P(Z_\gamma = z+1)}{P(Z_\gamma = z)} \right) - \frac{1}{\gamma^2} \left((z+1) \frac{P(Z_\gamma = z+1)}{P(Z_\gamma = z)} \right).$$

1353 Applying the quotient rule to the derivative inside the parentheses, we get

$$\frac{d}{d\gamma} \left(\frac{P(Z_\gamma = z+1)}{P(Z_\gamma = z)} \right) = \frac{P(Z_\gamma = z) \frac{d}{d\gamma} P(Z_\gamma = z+1) - P(Z_\gamma = z+1) \frac{d}{d\gamma} P(Z_\gamma = z)}{P(Z_\gamma = z)^2}.$$

1354 Using the properties of the Poisson distribution, specifically the identity

$$\frac{P(Z_\gamma = z+1)}{P(Z_\gamma = z)} = \frac{\gamma X}{z+1},$$

1355 we can simplify the derivative expression. Substituting back, we obtain

$$\frac{d}{d\gamma} E[X|Z_\gamma = z] = -z\gamma \text{Var}(X|Z_\gamma = z-1).$$

1356 For the case $z = 0$, the derivative simplifies to $\frac{d}{d\gamma} E[X|Z_\gamma = 0] = 0$, since $\text{Var}(X|Z_\gamma = -1) = 0$
 1357 by definition.

1358 The result for higher moments follows similarly. For any positive integer k , differentiating
 1359 $E[(\gamma X)^k|Z_\gamma = z]$ with respect to γ and applying the quotient rule leads to the stated piecewise
 1360 expression. This completes the proof.

1361 Moreover, for any positive integer k ,

$$\frac{d}{d\gamma} E[(\gamma X)^k|Z_\gamma = z] = \begin{cases} k E[(\gamma X)^{k-1}|Z_\gamma = 0], & z = 0, \\ \frac{(z+k) E[(\gamma X)^{k-1}|Z_\gamma = z] E[\gamma X|Z_\gamma = z-1] - z E[(\gamma X)^k|Z_\gamma = z]}{E[\gamma X|Z_\gamma = z-1]}, & z \geq 1. \end{cases}$$

1362 **Lemma 8.** Let $Z_\gamma \sim \mathcal{P}(\gamma X)$. Then, for every fixed $\gamma > 0$ and any non-degenerate X , the mapping
 1363 $z \mapsto E[X|Z_\gamma = z]$ is strictly increasing.

1364 *Proof.* To show that $E[X|Z_\gamma = z]$ is strictly increasing, we define $U = \gamma X$ and consider the Poisson
 1365 marginal probability:

$$P_{Z_\gamma}(k) = \frac{1}{k!} E[U^k e^{-U}] \quad (30)$$

1366 Applying the Cauchy-Schwarz inequality, we obtain

$$P_{Z_\gamma}(k) \leq \frac{1}{k!} \sqrt{E[U^{k+1} e^{-U}] E[U^{k-1} e^{-U}]}. \quad (31)$$

1367 Rewriting in terms of factorial expressions, we get

$$P_{Z_\gamma}(k) \leq \sqrt{\frac{k+1}{k}} P_{Z_\gamma}(k+1) P_{Z_\gamma}(k-1). \quad (32)$$

1368 Now, substituting this bound into the Turing-Good-Robbins (TGR) formula from Lemma 4:

$$E[U|Z_\gamma = z] = \frac{(z+1)P_{Z_\gamma}(z+1)}{P_{Z_\gamma}(z)}, \quad (33)$$

1369 we obtain the lower bound

$$E[U|Z_\gamma = z] \geq \frac{(z+1) \frac{z}{z+1} P_{Z_\gamma}^2(z)}{P_{Z_\gamma}(z) P_{Z_\gamma}(z-1)}. \quad (34)$$

1370 Simplifying, this reduces to

$$E[U|Z_\gamma = z] \geq \frac{z P_{Z_\gamma}(z)}{P_{Z_\gamma}(z-1)}. \quad (35)$$

1371 Using the same formulation for $z-1$, we conclude

$$E[U|Z_\gamma = z] \geq E[U|Z_\gamma = z-1]. \quad (36)$$

1372 Since $X = U/\gamma$, it follows that $E[X|Z_\gamma = z]$ is strictly increasing in z , completing the proof.

1373 **D.3 Incremental Channel Approach to I-MPRL and related proofs:**

1374 Here, we derive interesting relations between the mutual information in a Poisson noise channel and
 1375 various parameters of the channel. The general distribution we consider here is $Y \sim \text{Poisson}(\alpha X + \lambda)$.
 1376

1377 **Lemma 9.** Let $\lambda > 0$ and let X be a positive random variable satisfying $E\{X \log X\} < \infty$.
 1378 Consider the Poisson random transformation $X \mapsto Z_\lambda = \mathcal{P}(X + \lambda)$. Then, the derivative of the
 1379 mutual information between X and Z_λ with respect to the dark current λ is given by

$$\frac{d}{d\lambda} I(X; Z_\lambda) = E[\log(X + \lambda) - \log\langle X + \lambda \rangle],$$

1380 where $\langle X + \lambda \rangle = E[X + \lambda|Z_\lambda = z]$.

1381 *Proof:* Let $Y_0 = \mathcal{P}(X)$ and $N_\lambda = \mathcal{P}(\lambda)$ be independent Poisson random variables with means X
 1382 and λ , respectively. Define $Y_\lambda = Y_0 + N_\lambda$, which has the same distribution as $\mathcal{P}(X + \lambda)$. By the
 1383 definition of mutual information,

$$I(X; Y_0) - I(X; Y_\lambda) = E\{L(X, Y_0, Y_\lambda)\},$$

1384 where the expectation is over the joint distribution of (X, Y_0, Y_λ) , and the log-likelihood ratio is

$$L(x, k, \ell) = \log \frac{P_{Y_0|X}(k|x)}{P_{Y_0}(k)} - \log \frac{P_{Y_\lambda|X}(\ell|x)}{P_{Y_\lambda}(\ell)}.$$

1385 Given that $Y_0|X = x \sim \mathcal{P}(x)$ and $Y_\lambda|X = x \sim \mathcal{P}(x + \lambda)$, the conditional probabilities are

$$P_{Y_0|X}(k|x) = \frac{x^k e^{-x}}{k!}, \quad P_{Y_\lambda|X}(\ell|x) = \frac{(x + \lambda)^\ell e^{-(x + \lambda)}}{\ell!}.$$

1386 Substituting these into the log-likelihood ratio, we obtain

$$L(X, Y_0, Y_\lambda) = Y_0 \log X - Y_\lambda \log(X + \lambda) + U,$$

1387 where U encompasses terms involving the logarithms of the marginal probabilities. Taking the
1388 expectation, we have

$$E[L] = E\{X \log X - (X + \lambda) \log(X + \lambda)\} + E[U].$$

1389 Expanding $Y_\lambda = Y_0 + N_\lambda$ and leveraging the independence of N_λ from Y_0 , we analyze the behavior of
1390 $E[U]$ as λ becomes small. Through a series of manipulations and applying the dominated convergence
1391 theorem, we find that

$$I(X; Y_\lambda) - I(X; Y_0) = \lambda E \left[\log \frac{X}{\langle X \rangle} \right] + o(\lambda).$$

1392 Dividing both sides by λ and taking the limit as $\lambda \rightarrow 0$, we obtain

$$\frac{d}{d\lambda} I(X; Y_\lambda) = E [\log(X + \lambda) - \log \langle X + \lambda \rangle],$$

1393 where $\langle X + \lambda \rangle = E[X + \lambda | Y_\lambda = z]$. This completes the proof of Lemma 9.

1394 **Lemma 10.** For every Poisson transformation \mathcal{P}_X with $E\{X \log X\} < \infty$, and as $\delta \rightarrow 0$,

$$I(X; \mathcal{P}((1 + \delta)X)) - I(X; \mathcal{P}(X)) = \delta E\{X \log X - \langle X \rangle \log \langle X \rangle\} + o(\delta).$$

1395 *Proof:* Consider first the case $\delta \rightarrow 0^+$. Let $Y = \mathcal{P}(X)$ and $Z = \mathcal{P}(\delta X)$ be independent conditioned
1396 on X . Define $Y_\delta = Y + Z$. Then, the left-hand side of the lemma can be expressed as

$$I(X; Y_\delta) - I(X; Y) = E \left\{ \log \frac{P_{Y_\delta|X}(Y_\delta|X)}{P_{Y_\delta}(Y_\delta)} - \log \frac{P_{Y|X}(Y|X)}{P_Y(Y)} \right\}.$$

1397 Expanding the log-likelihood ratio, we have

$$= E \left\{ Z \log X - \delta X - \log \frac{E\{(X')^{Y_\delta} e^{-(1+\delta)X'} | Y_\delta\}}{E\{(X')^{Y_\delta} e^{-X'} | Y\}} \right\}.$$

1398 Here, X' is identically distributed as X but independent of Y and Z .

1399 To analyze the expression as $\delta \rightarrow 0$, we approximate $\Delta = \mathcal{P}(\delta X)$ by a Bernoulli random variable
1400 that takes the value 1 with probability δX (conditioned on X) and 0 otherwise. This approximation is
1401 valid because for small δ , the Poisson distribution $\mathcal{P}(\delta X)$ closely resembles a Bernoulli distribution.

1402 Substituting this approximation into the previous step, we obtain

$$I(X; Y_\delta) - I(X; Y) = E \left\{ Z \log X - \delta X - \log \left[(1 - \delta X) E\{(X')^Y e^{-X'} | Y\} + \delta X E\{(X')^{Y+1} e^{-X'} e^{-\delta X'} | Y\} \right] \right\} + o(\delta) \quad (37)$$

1403 Expanding $e^{-\delta X'}$ to first order in δ , we have $e^{-\delta X'} \approx 1 - \delta X'$. Therefore,

$$E\{(X')^{Y+1} e^{-X'} e^{-\delta X'} | Y\} \approx E\{(X')^{Y+1} e^{-X'} | Y\} - \delta E\{(X')^{Y+2} e^{-X'} | Y\} + o(\delta) \quad (38)$$

1404 Substituting this back into the logarithm and applying the first-order Taylor expansion $\log(1 + \epsilon) \approx \epsilon$
1405 for small ϵ , we obtain

$$\begin{aligned} & \log \left[(1 - \delta X) E\{(X')^Y e^{-X'} | Y\} + \delta X E\{(X')^{Y+1} e^{-X'} | Y\} \right] \\ & \approx \log \left[E\{(X')^Y e^{-X'} | Y\} \right] + \frac{\delta X E\{(X')^{Y+1} e^{-X'} | Y\} - \delta X E\{(X')^Y e^{-X'} | Y\}}{E\{(X')^Y e^{-X'} | Y\}} + o(\delta) \\ & = \log \langle X \rangle - \delta X \frac{E\{(X')^Y e^{-X'} | Y\} - E\{(X')^{Y+1} e^{-X'} | Y\}}{E\{(X')^Y e^{-X'} | Y\}} + o(\delta), \end{aligned}$$

1406 where $\langle X \rangle = E\{X | Y\}$.

1407 Substituting this approximation back into equation 37, we get

$$I(X; Y_\delta) - I(X; Y) = E \left\{ Z \log X - \delta X - \log \langle X \rangle + \delta X \frac{E\{(X')^{Y+1} e^{-X'} | Y\}}{E\{(X')^Y e^{-X'} | Y\}} \right\} + o(\delta) \quad (39)$$

1408 Noting that Z is Poisson with parameter X , we have $E\{Z|X\} = X$, and thus $E\{Z \log X\} =$
 1409 $E\{X \log X\}$.

1410 Furthermore, we know that $\langle X \rangle = E\{X|Y\}$, and from Lemma 4, we have

$$E\{(X')^{Y+1} e^{-X'} | Y\} = \langle X \rangle e^{-\langle X \rangle} (Y + 1).$$

1411 Substituting these into equation 39, we simplify to

$$I(X; Y_\delta) - I(X; Y) = \delta E\{X \log X - \langle X \rangle \log \langle X \rangle\} + o(\delta),$$

1412 Dividing both sides by δ and taking the limit as $\delta \rightarrow 0$, we obtain

$$\left. \frac{d}{d\delta} I(X; Y_\delta) \right|_{\delta=0} = E[X \log X - \langle X \rangle \log \langle X \rangle],$$

1413 where $\langle X \rangle = E[X|Y]$. This completes the proof of the lemma.

1414 E Tail Bounds

1415 As we know the output z_γ given the input x is modeled as $z_\gamma \sim \mathcal{P}(\gamma x)$, where $x \geq 0$ is the
 1416 non-negative input random variable, and γ represents the signal-to-noise ratio (SNR). The negative
 1417 log-likelihood when estimating z_γ using x , is given by:

$$l(x, z_\gamma) = -\log p(z_\gamma | x) = -\log \left(\frac{e^{-\gamma x} (\gamma x)^{z_\gamma}}{z_\gamma!} \right) = \gamma x - z_\gamma \log(\gamma x) + \log z_\gamma!$$

1418 We define the expected negative log-likelihood as $M(\gamma) = E_{(x, z_\gamma)}[l(x, z_\gamma)] = E_x[E_{(z_\gamma|x)}[l(x, z_\gamma)]]$.

1419 We now consider a mean constraint $\mu = E[x]$ in this case and our objective then is to determine the
 1420 input distribution $p_X(x)$ over $x \geq 0$ that maximizes the above function. To compute the expected
 1421 loss, let us first evaluate $E_{z_\gamma|x}[l(x, z_\gamma)]$ and using $E_{z_\gamma|x}[z_\gamma] = \gamma x$ gives:

$$E_{z_\gamma|x}[l(x, z_\gamma)] = E_{z_\gamma|x}[\gamma x - z_\gamma \log(\gamma x) + \log z_\gamma!] = \quad (40)$$

$$\gamma x - \log(\gamma x) \cdot E_{z_\gamma|x}[z_\gamma] + E_{z_\gamma|x}[\log z_\gamma!] = \gamma x - \gamma x \log(\gamma x) + E_{z_\gamma|x}[\log z_\gamma!] \quad (41)$$

1422 We can write $M(\gamma)$ in terms of the the conditional entropy of z_γ given x as:

$$M(\gamma) = E_x[H(z_\gamma|x)], \text{ since } H(z_\gamma|x) = E_{z_\gamma|x}[-\log p(z_\gamma|x)] = E_{z_\gamma|x}[l(x, z_\gamma)].$$

1423 The entropy $H(z_\gamma|x)$ of a Poisson distribution with parameter γx is given by:

$$HS(\gamma x) = - \sum_{k=0}^{\infty} P(z_\gamma = k) \log P(z_\gamma = k)$$

1424 where $P(z_\gamma = k) = \frac{(\gamma x)^k e^{-\gamma x}}{k!}$. So substituting this into the entropy expression, we obtain:

$$HS(\gamma x) = - \sum_{k=0}^{\infty} \frac{(\gamma x)^k e^{-\gamma x}}{k!} \log \left(\frac{(\gamma x)^k e^{-\gamma x}}{k!} \right) = \gamma x - \gamma x \log(\gamma x) + \sum_{k=0}^{\infty} \frac{(\gamma x)^k e^{-\gamma x}}{k!} \log k!$$

1425 It is natural to assume that the Shannon entropy $HS(\lambda)$ of a Poisson distribution strictly increases
 1426 with $\lambda \in (0, +\infty)$. We will prove this result, as well as the concavity property of $HS(\lambda)$, in the
 1427 following lemma.

1428 **Lemma 11.** *The Shannon entropy $HS(\lambda)$, $\lambda \in (0, +\infty)$, is strictly increasing and concave in λ .*

1429 *Proof.* The Shannon entropy $HS(\lambda)$ of a Poisson distribution is as outlined above. To analyze the
 1430 monotonicity and concavity of $HS(\lambda)$, we compute its first and second derivatives with respect to λ .
 1431 First, the first derivative $HS'(\lambda)$ is:

$$H'_S(\lambda) = -\log\left(\frac{\lambda}{e}\right) - 1 - e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^k \log k!}{k!} + e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^{k-1} \log k!}{(k-1)!} \quad (42)$$

$$= -\log \lambda + e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k \log(k+1)!}{k!} - e^{-\lambda} \sum_{k=2}^{\infty} \frac{\lambda^k \log k!}{k!} \quad (43)$$

1432 Simplifying, we get:

$$H'_S(\lambda) = -\log \lambda + e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k}{k!} \log(k+1)$$

1433 It is clear that both terms on the right-hand side of (2) are non-negative for $\lambda \in (0, 1]$, and the
 1434 second term is strictly positive. Therefore, $H'_S(\lambda) > 0$ for $\lambda \in (0, 1]$. Now, it remains to prove that
 1435 $H'_S(\lambda) > 0$ for $\lambda > 1$. Let's calculate:

$$\begin{aligned} H''_S(\lambda) &= -\frac{1}{\lambda} - e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k \log(k+1)}{k!} + e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^{k-1} \log(k+1)}{(k-1)!} \\ &= -\frac{1}{\lambda} + e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k \log(k+2)}{k!} - e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k \log(k+1)}{k!} \\ &= -\frac{1}{\lambda} + e^{-\lambda} \log 2 + e^{-\lambda} \sum_{k=1}^{\infty} \frac{\lambda^k \log\left(1 + \frac{1}{k+1}\right)}{k!} \\ &= -\frac{1}{\lambda} + e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k \log\left(1 + \frac{1}{k+1}\right)}{k!} \\ &< -\frac{1}{\lambda} + e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{(k+1)!} < -\frac{1}{\lambda} + e^{-\lambda} \frac{1}{\lambda} \sum_{k=0}^{\infty} \frac{\lambda^{k+1}}{(k+1)!} \\ &< -\frac{1}{\lambda} + e^{-\lambda} \frac{1}{\lambda} e^{\lambda} = 0. \end{aligned}$$

1436 So, $H''_S(\lambda) < 0$ for all $\lambda > 0$. Therefore, $H'_S(\lambda)$ strictly decreases in λ , proving **concavity** and it is
 1437 sufficient to prove that $\lim_{\lambda \rightarrow \infty} H'_S(\lambda) \geq 0$. After further simplification,

$$\lim_{\lambda \rightarrow \infty} H'_S(\lambda) = \lim_{\lambda \rightarrow \infty} \log \lambda \left(e^{-\lambda} (\log \lambda)^{-1} \sum_{k=1}^{\infty} \frac{\lambda^k \log(k+1)}{k!} - 1 \right),$$

1438 and it is sufficient to establish that

$$\liminf_{\lambda \rightarrow \infty} e^{-\lambda} (\log \lambda)^{-1} \sum_{k=1}^{\infty} \frac{\lambda^k \log(k+1)}{k!} \geq 1.$$

1439 This inequality is outlined in [58]. Using this, we get that $H'_S(\lambda) > 0$ for all $\lambda \geq 0$ and $H''_S(\lambda) < 0$
 1440 for all $\lambda \geq 0$, hence the proof follows.

1441 Given that $H(z_\gamma|x)$ is an increasing and concave function of x for $x > 0$, we aim to maximize
 1442 $E_x[H(z_\gamma|x)]$ under the mean constraint $E[x] = \mu$. The functional to maximize is $J[p_X(x)] =$
 1443 $\int_0^\infty H(z_\gamma|x) p_X(x) dx$, subject to the normalization and mean constraints: $\int_0^\infty p_X(x) dx =$
 1444 1 and $\int_0^\infty x p_X(x) dx = \mu$

1445 Introducing Lagrange multipliers λ and ν for these constraints, the Lagrangian becomes:

$$\mathcal{L}[p_X(x)] = \int_0^\infty H(z_\gamma|x)p_X(x) dx - \lambda \left(\int_0^\infty p_X(x) dx - 1 \right) - \nu \left(\int_0^\infty xp_X(x) dx - \mu \right)$$

1446 Taking the functional derivative of \mathcal{L} with respect to $p_X(x)$ and setting it to zero for optimality yields:

$$1447 \frac{\delta \mathcal{L}}{\delta p_X(x)} = H(z_\gamma|x) - \lambda - \nu x = 0$$

1448 Given the properties of $H(z_\gamma|x)$, the solution corresponds to an exponential distribution. The
1449 exponential distribution with mean μ is given by:

$$p_X(x) = \frac{1}{\mu} e^{-x/\mu}, \quad x \geq 0$$

1450 Maximizing the entropy of x leads to a distribution that spreads the probability mass, thereby
1451 increasing uncertainty and consequently maximizing the mprl. Now, using this exponential prior, we
1452 will derive an expression for $\text{mprl}(\gamma)$ which we use for deriving the left and right tail bounds.

1453 Now, the prior distribution for X is assumed to be an exponential distribution:

$$f_X(x) = \lambda e^{-\lambda x}$$

1454 We introduce the latent variable Z_γ such that:

$$P(Z_\gamma = z|X = x) = \frac{e^{-\gamma x}(\gamma x)^z}{z!}$$

1455 which follows a Poisson distribution. The conditional density of X given $Z_\gamma = z$ is derived as:

$$f_{X|Z}(x|z) = \frac{P(Z_\gamma = z|X = x)f_X(x)}{P(Z_\gamma = z)}$$

1456

$$f_{X|Z}(x|z) = \frac{(\beta x)^z}{z!} \lambda e^{-\lambda x} e^{-\beta x}$$

1457

$$= \frac{(\beta x)^z \lambda e^{-(\lambda+\beta)x}}{z! P(Z_\gamma = z)}$$

1458 and we can notice that this is a Gamma distribution: $X|Z_\gamma = z \sim \text{Gamma}(z+1, \lambda+\beta)$ The
1459 posterior mean of X given Z_γ is:

$$E[X|Z_\gamma = z] = \frac{z+1}{\lambda+\beta} \quad (44)$$

1460 and this serves as the optimal estimate \hat{X}^* . Now, let us consider the following expectation: (where l
1461 is the previously defined Poisson loss function)

$$E_{X|Z_\gamma} [l(X, X^*)] = E[X \log \left(\frac{X}{X^*} \right) - X + X^*] = E \left[X \log \left(\frac{X}{X^*} \right) \middle| Z_\gamma \right] - E[X|Z_\gamma] + X^* \quad (45)$$

1462 Using integration by parts and properties of the Gamma function, if $W \sim \text{Gamma}(\alpha, \beta)$, then: [59]

$$E[W \log W] = \frac{\alpha}{\beta} [\psi(\alpha+1) - \log \beta]$$

1463 where we defined the **digamma function** $\psi(\alpha)$ as: $\psi(\alpha) = \frac{d}{d\alpha} \log \Gamma(\alpha)$. The above results would
1464 also follow from differentiating the moment formula:

$$E[X^n] = \frac{\Gamma(\alpha+n)}{\Gamma(\alpha)\beta^n}$$

1465 Applying this this result in our case gives us:

$$E[X \log X | Z_\gamma] = \frac{z+1}{\lambda+\beta} [\psi(z+2) - \log(\lambda+\alpha)]$$

1466 We also have from Equation. 44:

$$\log(X^*) = \log(z+1) - \log(\lambda+\alpha)$$

1467 Taking expectation, the first term in Eq. 45 can be written as:

$$E \left[X \log \left(\frac{X}{X^*} \right) \middle| Z \right] = \frac{z+1}{\lambda+\beta} [\psi(z+2) - \log(\lambda+\alpha)] - \frac{z+1}{\lambda+\alpha} [\log(z+1) - \log(\lambda+\alpha)] \quad (46)$$

$$= \frac{z+1}{\lambda+\beta} [\psi(z+2) - \log(z+1)] \quad (47)$$

1468 Now, we compute the marginal distribution as follows:

$$P(Z_\gamma = z) = \int_0^\infty P(Z_\gamma = z | X = x) f_X(x) dx = \frac{\lambda \beta^z}{z!} \int_0^\infty x^z e^{-(\lambda+\beta)x} dx.$$

1469 Using the Gamma integral property stated as follows:

$$\int_0^\infty x^z e^{-(\lambda+\beta)x} dx = \frac{\Gamma(z+1)}{(\lambda+\beta)^{z+1}},$$

1470 we obtain (since $\Gamma(z+1) = z!$):

$$P(Z_\gamma = z) = \frac{\lambda \beta^z}{z!} \cdot \frac{\Gamma(z+1)}{(\lambda+\beta)^{z+1}} = \frac{\lambda \beta^z}{(\lambda+\beta)^{z+1}} = (1-p)p^z, \text{ where } p = \frac{\beta}{\lambda+\beta}$$

1471 Now, the $\text{mprl}(\gamma)$ expression obtained is as follows:

$$\text{mprl}(\gamma) = \sum_{z=0}^\infty (1-p)p^z \left[\frac{z+1}{\lambda+\beta} [\psi(z+2) - \log(z+1)] \right] = \frac{\lambda}{(\lambda+\beta)^2} \sum_{z=0}^\infty (z+1)p^z [\psi(z+2) - \log(z+1)].$$

1472 E.1 Left Tail Bound

1473 In case of (γ_0, γ_1) being the relevant range of integration, the left tail integral is defined as:

$$1474 \int_0^{\gamma_0} \text{mprl}(\gamma) d\gamma$$

1475 First, we interchange the sum and the integral:

$$\int_0^{\gamma_0} \text{mprl}(\gamma) d\gamma = \sum_{z=0}^\infty (z+1) [\psi(z+2) - \log(z+1)] \int_0^{\gamma_0} \frac{\lambda}{(\lambda+\gamma)^2} \left(\frac{\gamma}{\lambda+\gamma} \right)^z d\gamma.$$

1476 We define the inner integral as

$$I_z = \int_0^{\gamma_0} \frac{\lambda}{(\lambda+\gamma)^2} \left(\frac{\gamma}{\lambda+\gamma} \right)^z d\gamma.$$

1477 Substitute $u = \lambda + \gamma$, which implies $\gamma = u - \lambda$ and $d\gamma = du$. The bounds change accordingly:

1478 $u = \lambda$ when $\gamma = 0$ and $u = \lambda + \gamma_0$ when $\gamma = \gamma_0$. The integral becomes

$$I_z = \lambda \int_\lambda^{\lambda+\gamma_0} \frac{(u-\lambda)^z}{u^{z+2}} du.$$

1479 Next, using the substitution $v = \frac{u-\lambda}{u}$, leading to $u = \frac{\lambda}{1-v}$ and $du = \frac{\lambda}{(1-v)^2} dv$. The bounds

1480 transform to $v = 0$ when $u = \lambda$ and $v = \frac{\gamma_0}{\lambda+\gamma_0}$ when $u = \lambda + \gamma_0$. Substituting these into the integral

1481 yields

$$I_z = \int_0^{\frac{\gamma_0}{\lambda+\gamma_0}} v^z dv.$$

1482 The integral I_z can be evaluated as

$$I_z = \left[\frac{v^{z+1}}{z+1} \right]_0^{\frac{\gamma_0}{\lambda+\gamma_0}} = \frac{\left(\frac{\gamma_0}{\lambda+\gamma_0} \right)^{z+1}}{z+1}.$$

1483 Substituting I_z back into the expression for the expectation, gives:

$$\int_0^{\gamma_0} \text{mprl}(\gamma) d\gamma = \sum_{z=0}^{\infty} [\psi(z+2) - \log(z+1)] \left(\frac{\gamma_0}{\lambda+\gamma_0} \right)^{z+1}$$

1484 Let the above sum be S which we use in the sections below. By re-indexing the sum with $k = z + 1$,
1485 the final result can more elegantly be expressed as:

$$\int_0^{\gamma_0} \text{mprl}(\gamma) d\gamma = \sum_{k=1}^{\infty} [\psi(k+1) - \log(k)] \left(\frac{\gamma_0}{\lambda+\gamma_0} \right)^k.$$

1486 We aim to establish an upper bound for the sum

$$S = \sum_{z=0}^{\infty} (z+1) [\psi(z+2) - \log(z+1)] \left(\frac{\gamma_0}{\lambda+\gamma_0} \right)^{z+1},$$

1487 where ψ denotes the digamma function, $\gamma_0 > 0$, and $\lambda > 0$.

1488 Let us define $x = \frac{\gamma_0}{\lambda+\gamma_0}$. Given that $\gamma_0 > 0$ and $\lambda > 0$, it follows that $0 < x < 1$. From, [60], we
1489 recall the expansion of the digamma function:

$$\psi(z+2) = H_{z+1} - \gamma_E,$$

1490 where H_n is the n -th harmonic number and γ_E is the Euler-Mascheroni constant. For large z ,

$$H_{z+1} = \log(z+1) + \gamma_E + \frac{1}{2(z+1)} - \frac{1}{12(z+1)^2} + \cdots.$$

1491 Substituting this into the expression for $\psi(z+2)$ yields:

$$\psi(z+2) - \log(z+1) = \frac{1}{2(z+1)} - \frac{1}{12(z+1)^2} + \cdots.$$

1492 From this expansion, it is evident that

$$\psi(z+2) - \log(z+1) < \frac{1}{2(z+1)}$$

1493 for all $z \geq 0$, since the higher-order terms $-\frac{1}{12(z+1)^2} + \cdots$ contribute negatively, thereby decreasing
1494 the overall value.

1495 Consequently, each term in the sum satisfies

$$(z+1) [\psi(z+2) - \log(z+1)] x^{z+1} < \frac{1}{2} x^{z+1}.$$

1496 Summing over z from 0 to ∞ , we obtain

$$S < \frac{1}{2} \sum_{z=0}^{\infty} x^{z+1}.$$

1497 Using the simplification of the geometric series $\sum_{z=0}^{\infty} x^{z+1}$

$$\sum_{z=0}^{\infty} x^{z+1} = \frac{x}{1-x} \implies S < \frac{1}{2} \frac{x}{1-x}.$$

1498 Substituting back $x = \frac{\gamma_0}{\lambda+\gamma_0}$, we have

$$1-x = 1 - \frac{\gamma_0}{\lambda+\gamma_0} = \frac{\lambda}{\lambda+\gamma_0} \implies \frac{x}{1-x} = \frac{\frac{\gamma_0}{\lambda+\gamma_0}}{\frac{\lambda}{\lambda+\gamma_0}} = \frac{\gamma_0}{\lambda}.$$

1499 Putting this into the inequality for S , we obtain

$$S < \frac{1}{2} \frac{\gamma_0}{\lambda}.$$

1500 Hence, the upper bound for the sum in the scalar case (for a single input-output realization) is

$$\sum_{z=0}^{\infty} (z+1) [\psi(z+2) - \log(z+1)] \left(\frac{\gamma_0}{\lambda + \gamma_0} \right)^{z+1} \leq \frac{\gamma_0}{2\lambda}.$$

1501 (**Note:** This z is different from the z_γ notation used throughout the paper.)

1502 Extending this result to the vector case, consider a d -dimensional random vector $x \in X \subset \mathbb{Z}^d$ with
 1503 covariance matrix Σ , whose eigenvalues are $\{\lambda_i\}_{i=1}^d$, all positive. Assuming the problem is separable
 1504 across the eigenbasis of Σ , each dimension can be treated independently.

1505 For the vector case, the sum becomes

$$S_{\text{vector}} = \sum_{i=1}^d \sum_{z=0}^{\infty} (z+1) [\psi(z+2) - \log(z+1)] \left(\frac{\gamma_0}{\lambda_i + \gamma_0} \right)^{z+1}.$$

1506 Applying the scalar bound to each eigenvalue λ_i , we have

$$\sum_{z=0}^{\infty} (z+1) [\psi(z+2) - \log(z+1)] \left(\frac{\gamma_0}{\lambda_i + \gamma_0} \right)^{z+1} \leq \frac{\gamma_0}{2\lambda_i}.$$

1507 Summing over all i from 1 to d , the vector sum satisfies

$$S_{\text{vector}} \leq \sum_{i=1}^d \frac{\gamma_0}{2\lambda_i} = \frac{\gamma_0}{2} \sum_{i=1}^d \frac{1}{\lambda_i}.$$

1508 In the special case where the covariance matrix Σ is isotropic, meaning all eigenvalues $\lambda_i = \lambda$ for
 1509 $i = 1, \dots, d$, the bound simplifies to

$$S_{\text{vector}} \leq \frac{d\gamma_0}{2\lambda}.$$

1510 This concludes the derivation of the left tail bounds for both the scalar and vector cases.

1511 E.2 Right Tail Bound

1512 In case of (γ_0, γ_1) being the relevant range of integration, the right tail integral is defined as:
 1513 $\int_{\gamma_1}^{\infty} \text{mprl}(\gamma) d\gamma$

1514 Consider a discrete variable $x = (x_1, x_2, \dots, x_d) \in X \subset \mathbb{Z}^d$, where each component x_i belongs to a
 1515 discrete set $\{i\Delta | i \in \mathbb{Z}\}$. Observations are modeled as $z_{\gamma,i} \sim \mathcal{P}(\gamma x_i)$ for a large signal-to-noise ratio
 1516 (SNR) parameter γ . The estimator $\hat{x}_i(z_{\gamma,i})$ is typically the maximum likelihood estimator (MLE),
 1517 implemented by rounding $z_{\gamma,i}$ to the nearest bin $\{k\Delta\}$.

1518 The loss function per component is defined as

$$L(x_i, \hat{x}_i) = x_i \log \left(\frac{x_i}{\hat{x}_i} \right) - x_i + \hat{x}_i,$$

1519 and the $\text{mprl}(\gamma)$ is given by $\mathbb{E}[L(x_i, \hat{x}_i)]$ over the randomness of $z_{\gamma,i}$. The right-tail integral of
 1520 interest is

$$I_R = \int_{\gamma_1}^{\infty} E \left[\sum_{i=1}^d L(x_i, \hat{x}_i(z_{\gamma,i})) \right] d\gamma,$$

1521 which we aim to upper bound.

1522 At high SNR ($\gamma \rightarrow \infty$), the noise is relatively small compared to x_i , but rare rounding errors of size
 1523 $j\Delta$ can still occur. Focusing on a single component x_i , an error of size $j\Delta$ happens if

$$\hat{x}_i = x_i - j\Delta \iff z_{\gamma,i} \in [\gamma(x_i - j\Delta - 0.5\Delta), \gamma(x_i - j\Delta + 0.5\Delta)).$$

1524 For $z_{\gamma,i} \sim \text{Poisson}(\mu)$ with $\mu = \gamma x_i$, the Poisson Chernoff bound [61] provides that the probability
 1525 of such a deviation is at most $\exp(-c_{i,j}\gamma)$, where $c_{i,j} > 0$ is a constant dependent on Δ , x_i , and the
 1526 shift $j\Delta$. Hence,

$$P(\text{error of size } j\Delta) \leq e^{-c_{i,j}\gamma}.$$

1527 The per-component contribution to the mean MLE loss is

$$\text{mprl}_i(\gamma) = E_{z_{\gamma,i}} [L(x_i, \hat{x}_i(z_{\gamma,i}))].$$

1528 When the estimation error is $j\Delta$, the loss becomes

$$L(x_i, x_i - j\Delta) = x_i \log \left(\frac{x_i}{x_i - j\Delta} \right) - x_i + (x_i - j\Delta).$$

1529 Therefore, the mean loss satisfies

$$\text{mprl}_i(\gamma) \leq \sum_{j=1}^{j_{\max}} \left[x_i \log \left(\frac{x_i}{x_i - j\Delta} \right) - x_i + (x_i - j\Delta) \right] e^{-c_{i,j}\gamma}.$$

1530 Summing over all components $i = 1, \dots, d$, we obtain

$$\text{mprl}(\gamma) = \sum_{i=1}^d \text{mprl}_i(\gamma) \leq \sum_{i=1}^d \sum_{j=1}^{j_{\max}} \left[x_i \log \left(\frac{x_i}{x_i - j\Delta} \right) - x_i + (x_i - j\Delta) \right] e^{-c_{i,j}\gamma}.$$

1531 The right-tail integral I_R can thus be bounded as

$$I_R = \int_{\gamma_1}^{\infty} \text{mprl}(\gamma) d\gamma \leq \sum_{i=1}^d \sum_{j=1}^{j_{\max}} \left[x_i \log \left(\frac{x_i}{x_i - j\Delta} \right) - x_i + (x_i - j\Delta) \right] \int_{\gamma_1}^{\infty} e^{-c_{i,j}\gamma} d\gamma.$$

1532 Evaluating the integral, we find

$$\int_{\gamma_1}^{\infty} e^{-c_{i,j}\gamma} d\gamma = \frac{e^{-c_{i,j}\gamma_1}}{c_{i,j}},$$

1533 Leading to the final right-tail bound

$$I_R = \int_{\gamma_1}^{\infty} E \left[\sum_{i=1}^d L(x_i, \hat{x}_i) \right] d\gamma \leq \sum_{i=1}^d \sum_{j=1}^{j_{\max}} \left[x_i \log \left(\frac{x_i}{x_i - j\Delta} \right) - j\Delta \right] \frac{e^{-c_{i,j}\gamma_1}}{c_{i,j}}.$$

1534 In the above expression, $c_{i,j} > 0$ represents the Chernoff-type exponent from the Poisson large-
 1535 deviation bound for the event causing an error of size $j\Delta$ in component i . We determine these
 1536 parameters empirically, and the parameter j_{\max} indicates the largest error shift considered, which
 1537 is typically small in practice and can be tuned empirically. For empirical purposes, it might also be
 1538 worthwhile to note that the bracketed term in Eq. 47 can be approximated as the sum over a few
 starting z beyond which it effectively dies out as illustrated in Figure 14.

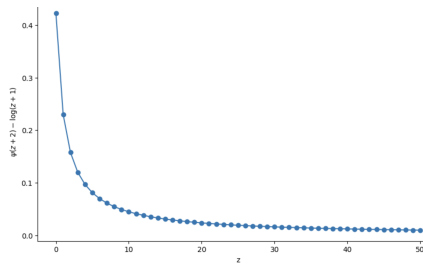


Figure 14: Approximating the Digamma term

1539

1540 **F Proof of Pointwise Poisson Denoising Relation**

1541 For Poisson channel observations $Z_\gamma \sim \text{Poisson}(\gamma X)$, we derive the pointwise denoising relation:

1542 **Lemma 12.** *The KL divergence derivative satisfies:*

$$\frac{d}{d\gamma} D_{KL}[P(z_\gamma|x) \| P(z_\gamma)] = \text{mprl}(x, \gamma)$$

1543 where the pointwise MPRL is:

$$\text{mprl}(x, \gamma) \equiv E_{P(z_\gamma|x)} [l(x, \hat{x}^*(z_\gamma))]$$

1544 with $l(x, x^*) = x \log \frac{x}{x^*} - x + x^*$ and $\hat{x}^*(z_\gamma) = E[X|z_\gamma]$.

1545 *Proof:* By definition,

$$D_{KL}(P(Z_\gamma|X) \| P(Z_\gamma)) = E_{P(x)} [E_{P(z_\gamma|x)} [\log P(z_\gamma|x) - \log P(z_\gamma)]] .$$

1546 We first move the differentiation inside expectations,

$$\frac{d}{d\gamma} D_{KL} = E_{P(x)} [E_{P(z_\gamma|x)} [\frac{d}{d\gamma} \log P(z_\gamma|x)]] - E_{P(x)} [E_{P(z_\gamma|x)} [\frac{d}{d\gamma} \log P(z_\gamma)]] .$$

1547 For the Poisson distribution with mean γx ,

$$P(z_\gamma|x) = e^{-\gamma x} \frac{(\gamma x)^{z_\gamma}}{z_\gamma!} .$$

1548 Taking the derivative,

$$\frac{d}{d\gamma} \log P(z_\gamma|x) = -x + \frac{z_\gamma}{\gamma} .$$

1549 Taking expectation w.r.t. $P(z_\gamma|x)$, the first term comes out to be zero.

$$E_{P(z_\gamma|x)} [-x + \frac{z_\gamma}{\gamma}] = -x + \frac{1}{\gamma} E_{P(z_\gamma|x)} [z_\gamma] = -x + \frac{\gamma x}{\gamma} = 0 .$$

1550 For the marginal,

$$P(z_\gamma) = \int P(z_\gamma|x) P(x) dx .$$

1551 Taking the log-derivative,

$$\frac{d}{d\gamma} \log P(z_\gamma) = \frac{1}{P(z_\gamma)} \int (-x + \frac{z_\gamma}{\gamma}) P(z_\gamma|x) P(x) dx .$$

1552 Identifying this as a conditional expectation,

$$\frac{d}{d\gamma} \log P(z_\gamma) = E[-X + \frac{z_\gamma}{\gamma} | z_\gamma] .$$

1553 Thus, the second term is

$$E_{P(x)} [E_{P(z_\gamma|x)} [\frac{d}{d\gamma} \log P(z_\gamma)]] = E_{P(z_\gamma|x)} [E[-X + \frac{z_\gamma}{\gamma} | z_\gamma]] .$$

1554 Combining both terms,

$$\frac{d}{d\gamma} D_{KL} = 0 - E_{P(z_\gamma|x)} [E[-X + \frac{z_\gamma}{\gamma} | z_\gamma]] .$$

1555

$$\frac{d}{d\gamma} D_{KL}(P(Z_\gamma|X) \| P(Z_\gamma)) = E_{P(z_\gamma|x)} [E[X|z_\gamma] - \frac{z_\gamma}{\gamma}] .$$

1556 **Link to the MPRL Loss:** We already defined the loss function:

$$\ell(x, \hat{x}^*) = x \log \frac{x}{\hat{x}^*} - x + \hat{x}^* .$$

1557 If $\hat{x}^* \equiv E[X|z_\gamma]$ is the estimator of x given z_γ , then by standard properties of conditional expectation,

$$E_{P(z_\gamma|x)}[\hat{x}^*] = E[E[X|z_\gamma]] = E[X] = x \quad (\text{if } x \text{ is deterministic, replace } E[X] \text{ by } x).$$

1558 Hence,

$$E_{P(z_\gamma|x)}[\ell(x, \hat{x}^*)] = E[x \log x - x \log \hat{x}^* - x + \hat{x}^*] = x \log x - x - xE[\log \hat{x}^*] + E[\hat{x}^*].$$

1559 Since $E[\hat{x}^*] = x$,

$$E_{P(z_\gamma|x)}[\ell(x, \hat{x}^*)] = x(\log x - E[\log \hat{x}^*]).$$

1560 One can show (by comparing with the final expression in the KL derivative) that this expectation
 1561 aligns with $E_{P(z_\gamma|x)}[E[X|z_\gamma] - \frac{z_\gamma}{\gamma}]$, thus establishing the link between the MPRL and the derivative
 1562 of the KL divergence. We can generalize this relation to any loss function that belongs to the class of
 1563 Bregman divergences in a Poisson channel using the framework described in [62].

1564 F.1 Tweedie's for Poisson Denoising

1565 A well-known result in Gaussian denoising is *Tweedie's Formula*, which expresses the conditional
 1566 expectation of the latent variable in terms of the derivative of the log-pdf of noisy observation. [31].
 1567 Specifically, for $Z_\gamma = \sqrt{\gamma}X + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$, we have:

$$E[X|Z_\gamma = z] = \frac{z}{\sqrt{\gamma}} + \frac{1}{\gamma} \nabla \log f_{Z_\gamma}(z), \quad (48)$$

1568 In the Poisson setting, we cannot directly take derivatives of $\log P_{Z_\gamma}(z)$ with respect to discrete z
 1569 since they are undefined. Instead, the *forward difference* of the log of the marginal PMF serves as a
 1570 discrete analog. This culminates in the Turing-Good-Robbins (TGR) formula, already presented in
 1571 Lemma 4.

1572 Hence, just like Tweedie's Formula in the continuous Gaussian case, TGR expresses the conditional
 1573 mean $\langle X \rangle_z$ purely in terms of the marginal distribution $P_{Z_\gamma}(z)$, bypassing any need to compute the
 1574 conditional distribution $P_{X|Z_\gamma}$. In effect, the ratio $\gamma \cdot \langle X \rangle_z$ plays the role of a *score function* for the
 1575 Poisson channel, analogous to the logarithmic derivative in the Gaussian case. This discrete variant
 1576 underpins our Poisson diffusion framework, allowing us to efficiently compute the optimal denoiser
 1577 $E[X|Z_\gamma]$ directly from the marginal PMF.

1578 G Continuous Extension of ItDPDM

1579 We extend the continuous-time channel with discrete states (CTDS) to continuous states through the
 1580 following construction:

1581 **Definition 1** (Continuous-Time Channel with States (CTCS)). *Let $\{X_t\}_{t \geq 0}$ be a right-continuous*
 1582 *state process with left limits (càdlàg) taking values in \mathbb{R}_+ . The output process $\{Y_t\}_{t \geq 0}$ is a counting*
 1583 *process satisfying:*

$$Y_t = \mathcal{P} \left(\int_0^t X_s ds \right) \quad (49)$$

1584 where $\mathcal{P}(\cdot)$ denotes a Poisson counting measure.

1585 For measurable intensity X_t , the output increments also satisfy:

$$Y_{t+\delta} - Y_t \sim \mathcal{P} \left(\int_t^{t+\delta} X_s ds \right), \quad \forall t, \delta \geq 0 \quad (50)$$

1586 with $\{Y_{t_k} - Y_{t_{k-1}}\}_{k=1}^n$ independent given $X_{[0,T]}$ for any finite partition $\{t_k\}$.

1587 The mutual information between state and observation processes over $[0, T]$ is given by:

$$I(X^T; Y^T) = E \left[\log \frac{dP_{Y^T|X^T}}{dP_{Y^T}} \right] \quad (51)$$

1588 The key connection to discrete-time systems emerges through infinitesimal discretization:

1589 **Lemma 13** (Mutual Information Rate). *For the CTCS in Definition 1, the mutual information rate*
 1590 *satisfies:*

$$\lim_{T \rightarrow \infty} \frac{1}{T} I(X^T; Y^T) = \lim_{\delta \rightarrow 0} \frac{1}{\delta} I(X_\delta; Y_\delta) \quad (52)$$

1591 where $X_\delta := X_{[0, \delta]}$ and $Y_\delta := Y_\delta - Y_0$ corresponds to the discrete-time channel $\mathcal{P}(\delta X)$.

1592 *Proof Sketch.* Consider time partitions $0 = t_0 < t_1 < \dots < t_n = T$ with $\max |t_{k+1} - t_k| \leq \delta$. By
 1593 the chain rule of mutual information:

$$\begin{aligned} I(X^T; Y^T) &= \sum_{k=0}^{n-1} I(X^{t_{k+1}}; Y_{t_{k+1}} | Y^{t_k}) \\ &= \sum_{k=0}^{n-1} [I(X_{[t_k, t_{k+1})}; Y_{[t_k, t_{k+1})}) + \epsilon_k] \end{aligned}$$

1594 where ϵ_k captures residual dependence between time intervals. Using the Markov property of Poisson
 1595 counters [63] and taking $\delta \rightarrow 0$, the residual terms vanish by the Asymptotic Equipartition Property
 1596 (AEP) for Poisson processes [64]. The result follows from Lemma 9 applied to each infinitesimal
 1597 interval.

1598 The continuous-time counterpart of the derivative relationship becomes:

1599 **Lemma 14** (Information Rate Derivative). *For the CTCS system, the time derivative of mutual*
 1600 *information satisfies:*

$$\frac{d}{dt} I(X^t; Y^t) = E[X_t \log X_t - \langle X_t \rangle \log \langle X_t \rangle] \quad (53)$$

1601 where $\langle X_t \rangle := E[X_t | Y^t]$ is the causal MPRL estimator.

1602 *Proof.* From Lemma 13 and the DTCS derivative, we have:

$$\begin{aligned} \frac{d}{dt} I(X^t; Y^t) &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [I(X_{t+\delta}; Y_{t+\delta} | Y^t) - I(X_t; Y_t)] \\ &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} E[\delta X_t \log X_t - \delta \langle X_t \rangle \log \langle X_t \rangle] + o(1) \end{aligned}$$

1603 The result follows by dominated convergence and the tower property of conditional expectation. This
 1604 continuous-time formulation preserves the essential duality between information and estimation seen
 1605 in discrete time, with the Poisson channel's inherent noise characteristics governing both regimes.
 1606 The CTCS framework enables analysis of real-time filtering and prediction [65] through differential
 1607 versions of the key discrete-time identities.

1608 H Detailed comparison of ItDPDM vs. Learning to Jump (LTJ, [20])

1609 Table 8 shows a detailed comparison below:

1610 In the Learning-to-Jump (LTJ) framework [20], the per-step training loss is written as $D_\phi(x, f_\theta(z_t, t))$,
 1611 where

- 1612 • $x \in \mathbb{N}$ is the true discrete count.
- 1613 • z_t is the noisy observation at step t , obtained by binomial thinning of z_{t-1} .
- 1614 • $f_\theta(z_t, t)$ is the denoising network (parameterized by θ), which takes (z_t, t) and outputs an
 1615 estimate \hat{x}_t of x .
- 1616 • $D_\phi(u, v)$ is the Bregman divergence induced by a convex generator ϕ : $D_\phi(u, v) =$
 1617 $\phi(u) - \phi(v) - \langle \nabla \phi(v), u - v \rangle$. For the Poisson channel one uses $\phi(u) = u \log u$, yielding
 1618 $D_\phi(x, \hat{x}) = \hat{x} \log \frac{\hat{x}}{x} - \hat{x} + x$, i.e. the Poisson–Bregman (relative-entropy) loss.

Table 8: Side-by-side comparison of our **ItDPDM** vs the **Learning-to-Jump** (LTJ) framework. We note that both methods employ a Poisson-Bregman (relative-entropy) loss—denoted PRL for ItDPDM and D_ϕ for LTJ but they diverge sharply in how that loss is used and how it connects to likelihood, as summarised below.

Aspect	ItDPDM (ours)	Learning-to-Jump (LTJ) [20]
Forward “noising”	Single-shot Poisson channel $Z_\gamma \sim \text{Pois}(\gamma X)$ with <i>continuous</i> SNR $\gamma \in (0, \infty)$	Binomial thinning chain $z_t \sim \text{Binomial}(z_{t-1}, \alpha_t / \alpha_{t-1})$ for $t = 1, \dots, T$
Reverse / generation	sampling operates in log-SNR space via a continuous-time reverse SDE or ODE; sampling can flexibly subsample the SNR continuum (e.g. 20–50 steps) without quality loss, in contrast to fixed-step chains	‘Count-thickening’ Markov chain with shifted-Poisson jumps; sampling requires executing all T discrete steps with no flexibility to skip or subsample, so the full T -step chain is incurred for every generated sample
Bounds on NLL	<i>Information-theoretic</i> , extends the classic I-MMSE identity to the Poisson channel, giving the exact relation: $-\log p(x) = \int_0^\infty \text{MPRL}(x, \gamma) d\gamma$	<i>Variational ELBO</i> , multi-term KL-divergence sum with binomial/Poisson factors; yields only an approximate bound on $-\log p(x)$
Training Loss	PRL : $\ell(x, \hat{x}) = \hat{x} \cdot \log(\hat{x}/x) - \hat{x} + x$, integrated over <i>continuous</i> γ , producing an exact NLL upper bound and provides analytic tail bounds & an importance-sampling estimator; empirically yields lower NLL than all baselines.	Per-step relative-entropy $D_\phi(x, f_\theta(z_t, t))$ inside an ELBO with an identical Bregman form, <i>but</i> summed over discrete T only with no closed-form link between the total loss and the true likelihood.
Scheduling	Choose only a continuous SNR grid (e.g., 1000-point logistic); no α_t or T hyper-parameters.	Must hand-design thinning schedule $\{\alpha_t\}_{t=1}^T$ and pick T (typically $T=1000$).
Likelihood evaluation	Exact tail bounds + importance sampling; likelihood (NLL) (in bits-per-dim) on real-world data, both WD & NLL on synthetic data evaluated	Likelihood <i>not</i> estimated; evaluation solely via Wasserstein distance (WD) of histograms.
Sampling speed	Compatible with fast ODE solvers (20–50 steps) due to continuous γ .	Must run all T thickening steps.
Theoretical extensions	Poisson-Tweedie identity; mutual-information derivative; CTCS extension.	—

I Noised and Denoised Image Comparison

Figure 19 presents a comparison of noisy and denoised images under Gaussian and Poisson noise conditions at a logSNR of 4.01. The left column displays the input images corrupted by Gaussian (Figure 17) and Poisson noise (Figure 15), while the right column shows the corresponding denoised outputs (Figures 16 and Figures 16). Notably, the Poisson noise case exhibits a higher level of degradation than the Gaussian noise case, making recovery more challenging. However, the denoising process effectively reconstructs meaningful image structures in both cases, demonstrating the model’s robustness to varying noise distributions.

J Theoretical Runtime Analysis of ItDPDM Architecture

We present a theoretical runtime analysis of the proposed *Information-Theoretic Discrete Poisson Diffusion Model* (ItDPDM), focusing on the core components contributing to its computational cost during training and inference.

Poisson Noise Sampling

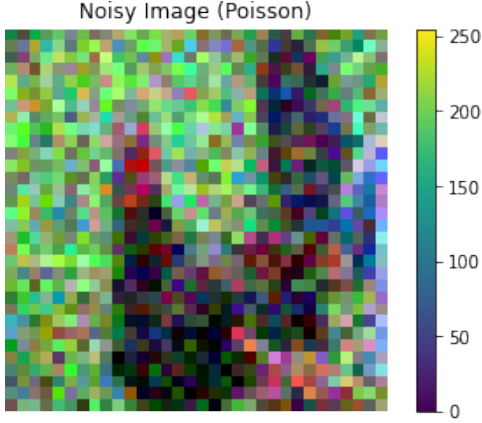


Figure 15: Noisy Image (Poisson Noise)
Noisy Image (Gaussian)

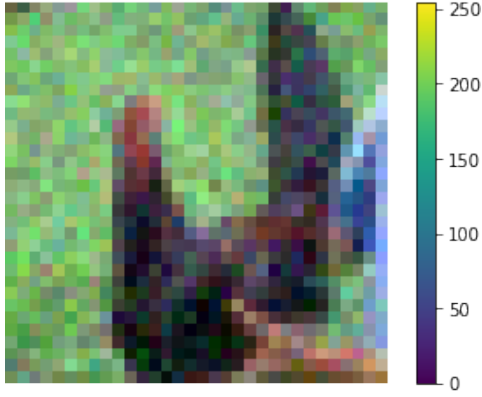


Figure 17: Noisy Image (Gaussian Noise)

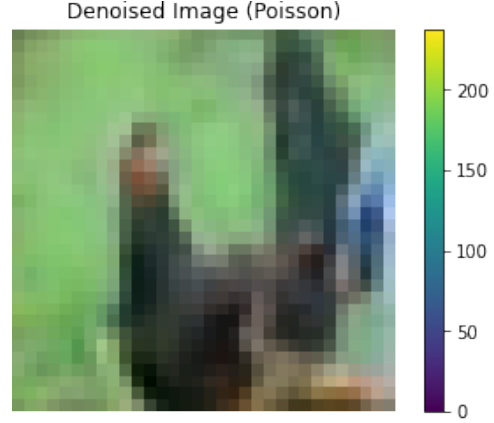


Figure 16: Denoised Image (Poisson Noise)
Denoised Image (Gaussian)

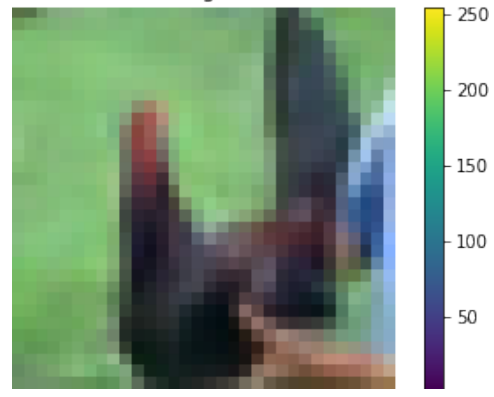


Figure 18: Denoised Image (Gaussian Noise)

Figure 19: Comparison of noisy and denoised images for poisoned and Gaussian noise birds with $\log\text{snr}=4.01$.

1632 The forward diffusion process in ItDPDM is governed by a Poisson noise channel $z_\gamma \sim \text{Poisson}(\gamma x)$,
 1633 where $x \in \mathbb{R}_+^D$ denotes the input data vector and γ is the signal-to-noise ratio (SNR). Sampling from
 1634 a Poisson distribution can be performed in $\mathcal{O}(1)$ per element using rejection sampling or table-based
 1635 methods, resulting in a total cost of $\mathcal{O}(D)$ per data point.

1636 **Neural Denoising**

1637 The denoiser is instantiated as a neural network, such as a U-Net (for images) or a Transformer
 1638 encoder (for symbolic music). The input to the denoiser is the reparameterized form

$$\tilde{z}_\gamma = \frac{z_\gamma}{1 + \gamma},$$

1639 which improves numerical stability. The forward pass of the denoiser has cost $\mathcal{O}(D)$ per data point,
 1640 assuming conventional convolutional or attention-based layers.

1641 **Poisson Loss Function Evaluation**

1642 The proposed loss function is based on a Bregman divergence tailored to Poisson noise:

$$\ell(x, \hat{x}) = x \log \left(\frac{x}{\hat{x}} \right) - x + \hat{x},$$

1643 which is convex, differentiable, and evaluated pointwise. The cost of loss evaluation and gradient
 1644 computation is $\mathcal{O}(D)$ per sample.

1645 **Integral Estimation over SNR**

1646 A defining component of the ItDPDM framework is the estimation of the negative log-likelihood
 1647 using thermodynamic integration:

$$-\log P(x) = \int_0^\infty \text{mprl}(x, \gamma) d\gamma,$$

1648 where MPRL denotes the minimum mean likelihood error. In practice, this integral is approximated
 1649 numerically using n log-SNR values (e.g., $n = 1000$), obtained via uniform or importance sampling
 1650 over $\alpha = \log \gamma$.

1651 Each SNR point requires a forward pass through the denoiser and loss computation, yielding a total
 1652 per-sample complexity of $\mathcal{O}(n \cdot D)$. To reduce overhead, the model uses importance sampling from a
 1653 truncated logistic distribution over α and closed-form tail integral bounds to truncate the SNR domain
 1654 (see Eqs. (28)–(29) in the main text).

Component	Complexity	Description
Poisson noise sampling	$\mathcal{O}(D)$	Efficient per-sample noise generation
Neural denoising	$\mathcal{O}(D)$	Forward pass through CNN or Transformer
Poisson loss function	$\mathcal{O}(D)$	Evaluated pointwise for each data coordinate
Integral over SNR	$\mathcal{O}(n \cdot D)$	Dominant cost due to repeated inference and loss evaluations
Total per-sample cost	$\mathcal{O}(n \cdot D)$	For fixed number of SNR grid points

Table 9: Asymptotic complexity of key components in the ItDPDM training pipeline.

1655 Given a batch size B and number of training epochs E , the overall training complexity becomes:
 1656 $\mathcal{O}(B \cdot E \cdot n \cdot D)$. This is comparable to standard continuous-state diffusion models using discretized
 1657 time steps, but the Poisson-specific formulation and MPRL integral introduce unique architectural
 1658 and optimization challenges that are efficiently addressed via reparameterization and sampling
 1659 strategies. Additionally, in terms of wall-clock times for training/sampling, we observe that ItDPDM
 1660 is comparable to standard DDPM-style models.

1661 K Extended Related Work

1662 Diffusion models have evolved along two orthogonal dimensions—*noise type* and *state space*.
 1663 Classical DDPMs corrupt continuous data with additive Gaussian noise and learn the reverse process
 1664 with score matching or variational bounds [2, 38, 46–48]. An information-theoretic viewpoint links
 1665 these objectives to mutual-information integrals [12, 21], and has recently motivated non-Gaussian
 1666 extensions based on annealed score matching [49, 50] and SDE formalisms [51]. Parallel work
 1667 seeks native *discrete-state* alternatives: masking schemes such as Blackout Diffusion employ an
 1668 irreversible “black” token that blocks exact likelihood computation [52]; Learning-to-Jump (LTJ)
 1669 replaces Gaussian noise by binomial thinning/thickening yet remains limited to discrete time and a
 1670 variational ELBO [20]. Very recent approaches move to continuous-time jump processes, but still
 1671 approximate the likelihood: [66] devise a categorical SDE whose reverse dynamics are learned by
 1672 discrete score matching, while [67] estimate probability *ratios* rather than scores to reduce perplexity
 1673 on text.

1674 **Score Entropy Discrete Diffusion (SEDD)** [67] represents a significant advancement in discrete
 1675 diffusion modeling. It introduces the *Score Entropy* loss, a novel objective that extends score matching
 1676 to discrete spaces by directly modeling the ratios of data probabilities. This approach addresses
 1677 the challenges of applying traditional score matching to discrete data and enables the construction
 1678 of discrete diffusion models that are both theoretically sound and empirically effective. SEDD
 1679 demonstrates competitive performance with autoregressive models like GPT-2 on standard language
 1680 modeling benchmarks. Notably, it achieves comparable zero-shot perplexities and offers advantages
 1681 in generation quality and efficiency. For instance, SEDD can generate high-quality text samples
 1682 with 4× lower generative perplexity when matching function evaluations and requires 16× fewer
 1683 function evaluations to match the generative perplexity of standard autoregressive sampling methods.
 1684 Moreover, SEDD enables arbitrary infilling beyond standard left-to-right prompting, matching the
 1685 quality of nucleus sampling without the need for specialized training or sampling techniques.

1686 Concurrently, several non-Gaussian *continuous* diffusion models have been proposed to address the
1687 limitations of traditional Gaussian-based approaches, particularly in handling data with bounded
1688 support or preserving structural details in images.

1689 **Beta Diffusion** [68] introduces a novel generative modeling method that integrates demasking and
1690 denoising to generate data within bounded ranges. Utilizing scaled and shifted beta distributions, it
1691 employs multiplicative transitions over time to create both forward and reverse diffusion processes.
1692 This approach maintains beta distributions in both the forward marginals and the reverse conditionals,
1693 given the data at any point in time. Unlike traditional diffusion models relying on additive Gaussian
1694 noise and reweighted evidence lower bounds (ELBOs), Beta Diffusion is multiplicative and optimized
1695 with KL-divergence upper bounds (KLUBs) derived from the convexity of the KL divergence.
1696 Experimental results demonstrate its unique capabilities in generative modeling of range-bounded
1697 data and validate the effectiveness of KLUBs in optimizing diffusion models.

1698 **Blurring Diffusion Models** [69] propose a generalized class of diffusion models that offer the best
1699 of both standard Gaussian denoising diffusion and inverse heat dissipation. By defining blurring
1700 through a Gaussian diffusion process with non-isotropic noise, this approach bridges the gap between
1701 inverse heat dissipation and denoising diffusion. It sheds light on the inductive bias resulting from this
1702 modeling choice and demonstrates the capability to better learn the low-to-mid frequencies within
1703 datasets, which plays a crucial role in representing shapes and structural information.

1704 **Edge-Preserving Noise** [70] for diffusion introduces a content-aware diffusion model explicitly
1705 trained to learn the non-isotropic edge information in a dataset. Inspired by anisotropic diffusion
1706 in image processing, this model incorporates an edge-aware noise scheduler that varies between
1707 edge-preserving and isotropic Gaussian noise. The generative process converges faster to results that
1708 more closely match the target distribution and better learns the low-to-mid frequencies within the
1709 dataset, crucial for representing shapes and structural information. This edge-preserving diffusion
1710 process consistently outperforms state-of-the-art baselines in unconditional image generation and
1711 is particularly robust for generative tasks guided by a shape-based prior, such as stroke-to-image
1712 generation

1713 While these models offer significant advancements in handling specific data characteristics, they still
1714 require dequantization and rely on surrogate objectives. In contrast, **ItDPDM** models corruption
1715 with a *reversible Poisson channel*, maintaining a discrete latent space, supporting bidirectional
1716 perturbations, and—via the I-MPRL identity—transforming the Minimum Poisson Reconstruction
1717 Loss into an *exact* likelihood integral instead of a bound. This unifies the tractability of information-
1718 theoretic Gaussian diffusion with the fidelity of discrete-state models, yielding closed-form NLLs,
1719 scalable continuous-time sampling, and strong empirical performance on sparse, skewed, and over-
1720 dispersed count data

1721 **ItDPDM** differs fundamentally from the above lines. By modelling corruption with a *reversible*
1722 *Poisson channel*, ItDPDM keeps the latent space discrete, supports bidirectional perturbations,
1723 and—via the I-MPRL identity—turns the Minimum Poisson Reconstruction Loss into an *exact*
1724 likelihood integral instead of a bound. This unifies the tractability of information-theoretic Gaussian
1725 diffusion with the fidelity of discrete-state models, yielding closed-form NLLs, scalable continuous-
1726 time sampling, and strong empirical performance on sparse, skewed, and over-dispersed count
1727 data.