

---

# Conditional Diffusion Probabilistic Models for Super Resolution Microscopy

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Single-molecule localization microscopy (SMLM) techniques are a mainstay of  
2 fluorescence microscopy and can be used to produce a pointillist representation of  
3 living cells at diffraction-unlimited precision. Classical SMLM approaches lever-  
4 age the deactivation of fluorescent tags, followed by spontaneous or photoinduced  
5 reactivation, which can be used to estimate of the density of a tagged biomolecule  
6 in cellular compartments. Standard SMLM localization algorithms based on max-  
7 imum likelihood estimators or least squares optimization require tight control  
8 of activation and reactivation to maintain sparse emitters, presenting a tradeoff  
9 between imaging speed and labeling density. Recently, deep models have gener-  
10 alized SMLM to densely labeled structures by predicting high-resolution kernel  
11 density estimates (KDEs) from low resolution images with convolutional networks.  
12 However, estimated KDEs may contain irregularities due to finite sample sizes  
13 and limited model capacity. Denoising diffusion probabilistic models (DDPMs) are  
14 well suited conditional super resolution tasks, demonstrating promising results in  
15 detail reconstruction, while directly providing uncertainties in model predictions.  
16 Here, we adapt DDPM to the task of single molecule localization, and demonstrate  
17 that combining traditional CNNs with a DDPM permits uncertainty quantification  
18 of KDEs and improves localization precision over a wide range of experimental  
19 conditions.

## 20 1 Introduction

21 Single molecule localization microscopy (SMLM) relies on the temporal resolution of fluorophores  
22 whose spatially overlapping point spread functions would otherwise render them unresolvable  
23 at the detector. Common strategies for the temporal separation of molecules involve molecular  
24 photoswitching from dark to fluorescent. Estimation of molecular coordinates is then carried out via  
25 modeling the optical impulse response of the imaging system and fitting model functions to the data.  
26 However, such models are only well-suited to isolated molecules, reducing the number of molecules  
27 in the field of view and limiting temporal resolution in super resolution microscopy. This issue has  
28 incited a series of efforts to increase the density of fluorescent molecules imaged in a single frame  
29 while developing appropriate models for dense localization.

30 Previous approaches to this issue has been to predict super-resolution images from a sparse set of  
31 localizations with conditional generative adversarial networks (Ouyang 2018) or direct prediction  
32 of molecular coordinates using neural networks (Nehme 2020; Speiser 2021). However, diffusion  
33 models are an appealing alternative because they model a distribution of high-resolution images that  
34 are compatible with a measurement. Although conditional VAEs and conditional GANs can provide  
35 a distribution of images with enhanced resolution, both are known to suffer from mode collapse and  
36 produce insufficient diversity in their outputs. Diffusion models are a recently developed alternative

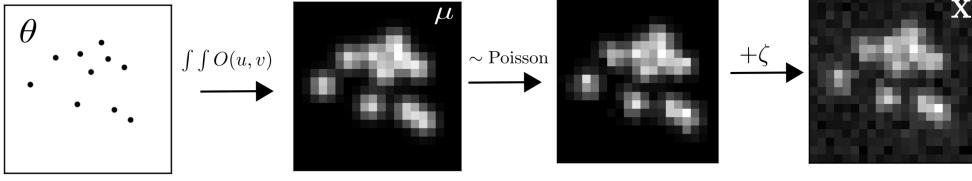


Figure 1: Generative model of single molecule localization microscopy images

to VAEs and GANs that excel at producing diverse samples and have been successfully applied to solve inverse problems.

Here we focus on the class of models which perform single molecule localization using neural networks. In this approach, one estimates molecular coordinates by predicting kernel density estimates (KDEs)  $\mathbf{y}$ , which are latent in the raw data  $\mathbf{x}$ , using a convolutional neural network DeepSTORM, followed by thresholding (Nehme 2021). Such methods are currently the state of the art for dense localization microscopy, but may exhibit localization bias, and produce KDEs with aberrant structure due to lack of regularization. Building on this work, we propose combining a modified DeepSTORM architecture with a denoising diffusion probabilistic model (DDPM) which models a distribution of KDEs  $\mathbf{y}$ , providing a novel mechanism for uncertainty quantification.

## 2 Background

### 2.1 Image Degradation Model

The central objective of single molecule localization microscopy is to infer a set of molecular coordinates  $\theta$  from noisy, low resolution images  $\mathbf{x}$ . We therefore begin by defining the likelihood on measured low-resolution images  $p(\mathbf{x}|\theta)$ . In fluorescence microscopy, each pixel is a Poisson random variable (Smith 2010; Nehme 2020; Chao 2016), with expected value

$$\omega = i_0 \int O(u) du \int O(v) dv \quad (1)$$

where  $i_0 = \eta N_0 \Delta$ . The scalar parameters  $\eta, \Delta$  are the photon detection probability of the sensor and the exposure time, respectively. Without loss of generality, we assume  $\eta = \Delta = 1$ . Most importantly,  $N_0$  represents the signal amplitude, which we assume maintains a fixed value. The optical impulse response  $O(u, v)$  is often approximated as a 2D isotropic Gaussian with standard deviation  $\sigma$  (Zhang 2007). This approximation has the convenient property, that the effects of pixelation can be expressed in terms of error functions. For example, given a fluorescent emitter located at  $\theta = (u_0, v_0)$ , we have that

$$\int O(u) du = \frac{1}{2} \left( \operatorname{erf} \left( \frac{u_k + \frac{1}{2} - u_0}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left( \frac{u_k - \frac{1}{2} - u_0}{\sqrt{2}\sigma} \right) \right) \quad (2)$$

where we have used the common definition  $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-t^2} dt$ . Our generative model also incorporates a normally distributed white noise per pixel  $\zeta$  with offset  $o$  and variance  $\sigma^2$ . Ultimately, we have a Poisson component of the signal, which scales with  $N_0$  and a Gaussian component, which does not. Therefore, in a single exposure, we measure:

$$\mathbf{x} = \mathbf{s} + \zeta \quad (3)$$

The distribution of  $\mathbf{x}$  is the convolution of the distributions of  $\mathbf{s}$  and  $\zeta$ ,

$$p(\mathbf{x}_k|\theta) = A \sum_{q=0}^{\infty} \frac{1}{q!} e^{-\omega_k} \omega_k^q \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(\mathbf{x}_k - g_k q - o_k)^2}{2\sigma_k^2}} \quad (4)$$

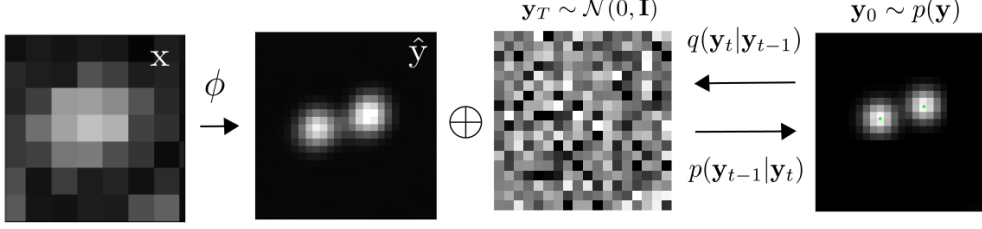


Figure 2: Conditional diffusion model for sampling kernel density estimates

where  $p(\zeta_k) = \mathcal{N}(o_k, \sigma_k^2)$  and  $p(s_k) = \text{Poisson}(\omega_k)$ ,  $A$  is some normalization constant. In practice, (4) is difficult to work with, so we look for an approximation. We will use a Poisson-Normal approximation for simplification. Consider,

$$\zeta_k - o_k + \sigma_k^2 \sim \mathcal{N}(\sigma_k^2, \sigma_k^2) \approx \text{Poisson}(\sigma_k^2) \quad (5)$$

Since  $\mathbf{x}_k = \mathbf{s}_k + \zeta_k$ , we transform  $\mathbf{x}'_k = \mathbf{x}_k - o_k + \sigma_k^2$ , which is distributed according to

$$\mathbf{x}'_k \sim \text{Poisson}(\omega'_k) \quad (6)$$

where  $\omega'_k = \omega_k + \sigma_k^2$ . This result can be seen from the fact the the convolution of two Poisson distributions is also Poisson. We then arrive at the following log likelihood

$$\ell(\mathbf{x}|\theta) = -\log \prod_k \frac{e^{-(\mu'_k)} (\mu'_k)^{n_k}}{n_k!} \approx \sum_k n_k \log n_k + \mu'_k - n_k \log (\mu'_k) \quad (7)$$

### 3 Denoising Diffusion Probabilistic Model for SMLM

We consider datasets  $(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{y}}_i)_{i=1}^N$  of observed images  $\mathbf{x}_i$  true kernel density estimate (KDE) images  $\mathbf{y}_i$ , and KDE estimates  $\hat{\mathbf{y}}_i = \phi(\mathbf{x}_i)$ . Observations  $\mathbf{x}_i$  are generated under the image degradation model. We aim to develop a framework for sampling from  $p(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{y})$ .

### 4 Conditional Denoising Diffusion Model

Point estimates  $\hat{\mathbf{y}}_i$  produced by the DeepSTORM architecture lack uncertainty quantification. To address this, we propose a DDPM to model the conditional distribution  $p(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{y})$ . Consider the factorization  $p(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{y})p(\mathbf{x}|\mathbf{y})p(\mathbf{y}) = p(\mathbf{x}|\mathbf{y}, \hat{\mathbf{y}})p(\mathbf{y}|\hat{\mathbf{y}})p(\hat{\mathbf{y}})$ . Given that  $\mathbf{x}$  is conditionally independent of  $\hat{\mathbf{y}}$ , we find

$$p_\Psi(\hat{\mathbf{y}}|\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\hat{\mathbf{y}})$$

where the DDPM  $\Psi$  is trained on pairs  $(\mathbf{y}_i, \hat{\mathbf{y}}_i)_{i=1}^N$ . The conditional DDPM generates a target image  $\mathbf{y}_0$  in  $T$  refinement steps. Starting with a pure noise image  $\mathbf{y}_T \sim \mathcal{N}(0, \mathbf{I})$ , the model iteratively refines the image through successive iterations according to learned conditional transition distributions  $p(\mathbf{y}_{t-1}|\mathbf{y}_t, \cdot)$  such that  $\mathbf{y}_0 \sim p(\mathbf{y}|\hat{\mathbf{y}})$

#### 4.1 Gaussian Diffusion

Diffusion models (Sohl-Dickstein 2015; Ho 2020) are a class of generative models inspired by nonequilibrium statistical physics, which slowly destroy structure in a data distribution  $p(\mathbf{y}_0|\mathbf{x})$  via a fixed Markov chain referred to as the *forward process*. In essence, the forward process gradually adds Gaussian noise to the data according to a variance schedule  $\beta_{0:T}$

$$q(\mathbf{y}_t|\mathbf{y}_0) = \prod_{t=1}^T q(\mathbf{y}_t|\mathbf{y}_{t-1}) \quad q(\mathbf{y}_t|\mathbf{y}_{t-1}) = \mathcal{N}\left(\sqrt{1-\beta_t}\mathbf{y}_{t-1}, \beta_t I\right) \quad (8)$$

89 An important property of the forward process is that it admits sampling  $x_t$  at an arbitrary timestep  $t$   
 90 in closed form (Ho 2020). Using the notation  $\alpha_t := 1 - \beta_t$  and  $\gamma_t := \prod_{s=1}^t \alpha_s$ , we have

$$q(\mathbf{y}_t|\mathbf{y}_0) = \mathcal{N}(\sqrt{\gamma_t}\mathbf{y}_0, (1 - \gamma_t)I) \quad (9)$$

91 The usual procedure is then to learn a parametric representation of the *reverse process*, and therefore  
 92 generate samples from  $p(\mathbf{y}_0)$ , starting from noise. Here, we are concerned with conditional diffusion  
 93 models, which sample from a distribution  $p(\mathbf{y}_0|\hat{\mathbf{y}})$ . Formally,  $p_\theta(\mathbf{y}_0|\hat{\mathbf{y}}) = \int p_\theta(\mathbf{y}_{0:T}|\mathbf{x}_0)d\hat{\mathbf{y}}_{1:T}$   
 94 where  $\mathbf{y}_t$  is a latent representation with the same dimensionality of the data.  $p_\theta(\mathbf{y}_{0:T}|\hat{\mathbf{y}})$  is a Markov  
 95 process, starting from a noise sample  $p_\theta(\mathbf{y}_T) = \mathcal{N}(0, \mathbf{I})$ .

$$p_\theta(\mathbf{y}_{0:T}) = p_\theta(\mathbf{y}_T) \prod_{t=1}^T p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t) \quad p_\theta(\mathbf{y}_{t-1}|\mathbf{y}_t) = \mathcal{N}(\mu_\theta(\mathbf{y}_t), \beta_t I) \quad (10)$$

96 where we reuse the variance schedule of the forward process (Ho 2020). We seek to learn a denoising  
 97 model  $\mu_\theta$  which computes the mean of the Gaussian transition density at each time step  $t$ . However,  
 98 learning diffusion models directly in data space can limit expressivity of the model (Vahdat 2021).  
 99 Since we are primarily interested in learning a restoration  $\mathbf{y}$ , we choose to define an encoder  $\phi$  such  
 100 that  $\mathbf{z} = \phi(\mathbf{x}_0)$ . The reverse process then becomes  $p_\theta(\mathbf{y}_0|\mathbf{z} = \phi(\mathbf{x}_0)) = \int p_\theta(\mathbf{y}_{0:T}|\mathbf{z})d\mathbf{x}_{1:T}$ . For all  
 101  $t > 0$ , the mean of the transition density is computed as

$$\mu_\theta(\mathbf{y}_t, \hat{\mathbf{y}}, \gamma_t) = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{y}_t - \frac{(1 - \alpha_t)}{\sqrt{1 - \gamma_t}} f_\theta(\mathbf{y}, \hat{\mathbf{y}}, \gamma_t) \right) \quad (11)$$

102 where  $f_\theta$  is a neural network. Only at  $t = 0$  is this mean directly a function of  $\mathbf{x}$ .

## 103 4.2 Optimization of the Denoising Model

104 To reverse the diffusion process, we optimize a neural denoising model  $f_\theta$  that takes as input  $\hat{\mathbf{y}}$  and a  
 105 noisy target image  $\mathbf{y}_t \sim q(\mathbf{y}_t|\mathbf{y}_0)$ ,

$$\mathbf{y}_t = \sqrt{\gamma}\mathbf{y}_0 + \sqrt{1 - \gamma}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (12)$$

106 This definition of a noisy target image  $\mathbf{y}_t$  is drawn from the marginal distribution of noisy images at  
 107 a time step  $t$  of the forward diffusion process. In addition to a source image  $\mathbf{y}_0$  and a noisy target  
 108 image  $\mathbf{y}_t$ , the denoising model  $f_\theta$  takes as input the sufficient statistics for the variance of the noise  
 109  $\gamma$ , and is trained to predict the noise vector  $\epsilon$ . We make the denoising model aware of the level of  
 110 noise through conditioning on a scalar  $\gamma$ . The proposed objective function for training  $f_\theta$  is

$$\mathbb{E}_{(\hat{\mathbf{y}}, \mathbf{y}_0)} \mathbb{E}_{(\epsilon, \gamma)} \left[ f_\theta \left( x, \sqrt{\gamma}\mathbf{y}_0 + \sqrt{1 - \gamma}\epsilon \mid \mathbf{y}_t, \gamma \right) - \epsilon \right], \quad (13)$$

111 where  $(\hat{\mathbf{y}}, \mathbf{y}_0)$  is sampled from the training dataset and  $\gamma \sim p(\gamma)$ . The distribution of  $\gamma$  has a big  
 112 impact on the quality of the model and the generated outputs. For our training noise schedule, we  
 113 use a piecewise distribution for  $\gamma$ ,  $p(\gamma) = \frac{1}{T} \sum_{t=1}^T U(\gamma_{t-1}, \gamma_t)$  (Nanxin 2021). Specifically, during  
 114 training, we first uniformly sample a time step  $t \sim \{0, \dots, T\}$  followed by sampling  $\gamma \sim U(\gamma_{t-1}, \gamma_t)$ .  
 115 We set  $T = 100$  in all our experiments.

## 116 4.3 Optimization of the DeepSTORM encoder

117 A first pass at localization treats localization as a binary classification problem, such that 0 denotes  
 118 a vacant pixel and 1 denotes an occupied pixel containing an emitter. Direct learning of pixel-wise

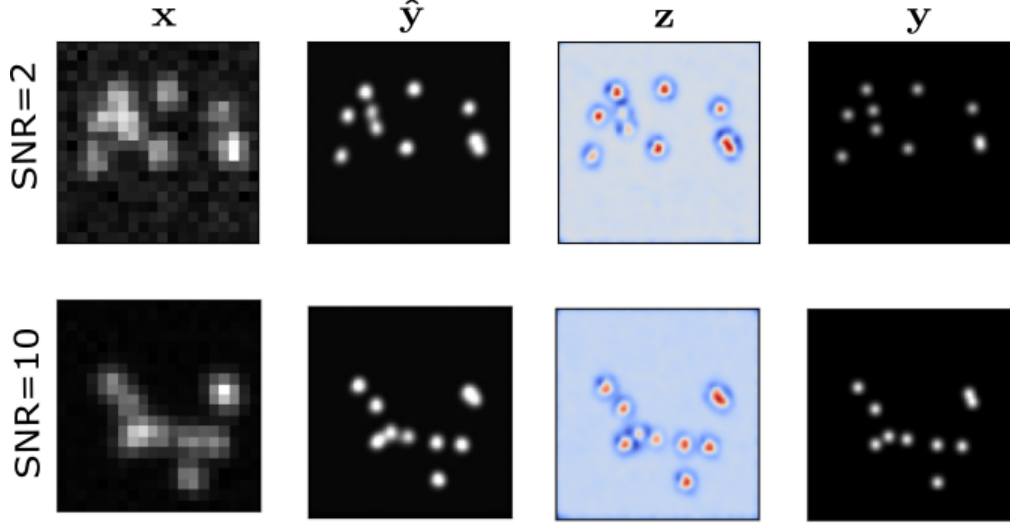


Figure 3: Kernel density estimates for various signal to noise ratios (SNR)

classification with cross-entropy loss leads to an imbalance of occupied and unoccupied pixels in dense localization problems (Nehme 2020). CE loss is usually either weighted [51], replaced with a Focal loss [52], or applied to a "blobbed" version of the desired boolean volume e.g. by placing a disk around each GT position [53–55]. Alternative methods take a soft version of the binary classification problem. That is, by placing a small Gaussian around each GT position (e.g. with std of 1 pixel), and matching continuous heatmaps, backpropagation yields more meaningful gradients and eases the learning process convergence.

Localization heatmaps thus form a natural encoding for SMLM images, which can be input to our conditional diffusion model. Therefore, to encode raw data  $\mathbf{x}$  into a more tractable representation, we train the DeepSTORM architecture (Nehme 2020). Raw coordinates  $\theta$  are binned into an upsampled image  $\mathbf{z}$ .

$$\mathcal{L}(\mathbf{z}, \hat{\mathbf{z}}) = \|\mathbf{z} * K - \hat{\mathbf{z}} * K\|^2 \quad (14)$$

## 5 Experiments

We set  $T = 100$  for all experiments and treat forward process variances  $\beta_t$  as hyperparameters, with a linear schedule from  $\beta_0 = 10^{-4}$  to  $\beta_T = 10^{-2}$ . These constants were chosen to be small relative to data scaled to  $[-1, 1]$ , ensuring that reverse and forward processes have approximately the same functional form while keeping the signal-to-noise ratio at  $x_T$  as small as possible ( $L_T = D_{KL}(q(x_T|x_0) \parallel \mathcal{N}(0, I)) \approx 10^{-5}$  bits per dimension in our experiments).

To represent the reverse process, we used the DDPM architecture based on a U-Net backbone (Ho 2020). Parameters are shared across time, which is specified to the network using the Transformer sinusoidal position embedding ?. We use self-attention at the  $16 \times 16$  feature map resolution ?. Details are in Appendix A.

and the channel multipliers at different resolutions (see Appendix A for details). To condition the model on the input  $x$ , we up-sample the low-resolution image to the target resolution using bicubic interpolation. The result is concatenated with  $y_t$  along the channel dimension. We experimented with more sophisticated methods of conditioning, such as using, but we found that the simple concatenation yielded similar generation quality.

## 145 5.1 Localization Error Analysis

## 146 6 Related Work

### 147 6.1 Diffusion Models

148 Prior work of diffusion models ?? require 1-2k diffusion steps during inference, making generation  
149 slow for large target resolution tasks. We adapt techniques from ? to enable more efficient inference.  
150 Our model conditions on  $\gamma$  directly (vs  $t$  as in ?), which allows us flexibility in choosing the number  
151 of diffusion steps, and the noise schedule during inference. This has been demonstrated to work  
152 well for speech synthesis ?, but has not been explored for images. For efficient inference, we set the  
153 maximum inference budget to 100 diffusion steps, and hyper-parameter search over the inference  
154 noise schedule. This search is inexpensive as we only need to train the model once ?. We use FID on  
155 held-out data to choose the best noise schedule, as we found PSNR did not correlate well with image  
156 quality.

### 157 6.2 Localization Microscopy with Deep Networks

### 158 6.3 Fisher Information Metric

159 We use the Fisher information as an information theoretic criteria to assess the quality of the proposed  
160 algorithms, with respect to the root mean squared error (RMSE) of our predictions of  $\theta$ . The  
161 generative model  $\ell(\mathbf{x}|\theta)$  is also convenient for computing the Fisher information matrix (Smith  
162 2010) and thus the Cramer-Rao lower bound, which bounds the variance of a statistical estimator  
163 of  $\theta$ , from below i.e.,  $\text{var}(\hat{\theta}) \geq I^{-1}(\theta)$ . It is shown in the appendix, that the Fisher information is  
164 straightforward to compute under the Poisson likelihood (7)

$$\mathcal{I}_{ij}(\theta) = \mathbb{E}_{\theta} \left( \frac{\partial \ell}{\partial \theta_i} \frac{\partial \ell}{\partial \theta_j} \right) = \sum_k \frac{1}{\omega'_k} \frac{\partial \omega'_k}{\partial \theta_i} \frac{\partial \omega'_k}{\partial \theta_j} \quad (15)$$