

DiffSED: Sound Event Detection with Denoising Diffusion

Swapnil Bhosale^{1*}, Sauradip Nag^{1*}, Diptesh Kanodia¹, Jiankang Deng², Xiatian Zhu¹

¹University of Surrey, UK

²Imperial College London, UK

s.bhosale@surrey.ac.uk*, s.nag@surrey.ac.uk*

Abstract

Sound Event Detection (SED) aims to predict the temporal boundaries of all the events of interest and their class labels, given an unconstrained audio sample. Taking either the split-and-classify (*i.e.*, frame-level) strategy or the more principled event-level modeling approach, all existing methods consider the SED problem from the discriminative learning perspective. In this work, we reformulate the SED problem by taking a generative learning perspective. Specifically, we aim to generate sound temporal boundaries from noisy proposals in a denoising diffusion process, conditioned on a target audio sample. During training, our model learns to reverse the noising process by converting noisy latent queries to the ground-truth versions in the elegant Transformer decoder framework. Doing so enables the model to generate accurate event boundaries from even noisy queries during inference. Extensive experiments on the Urban-SED and EPIC-Sounds datasets demonstrate that our model significantly outperforms existing alternatives, with 40% faster convergence in training. Code: <https://github.com/Surrey-UPLab/DiffSED>.

Introduction

Sound event detection (SED) aims to temporally localize sound events of interest (*i.e.*, the start and end time) and recognize their class labels in a long audio stream (Mesaros et al. 2021). As a fundamental audio signal processing task, it has become the cornerstone of many related recognition scenarios, such as audio captioning (Xu et al. 2021; Bhosale, Chakraborty, and Kopparapu 2023; Xie et al. 2023), and acoustic scene understanding (Igarashi et al. 2022; Bear, Nolasco, and Benetos 2019).

In the literature, all existing SED methods can be grouped into two categories namely, frame-level and event-level approaches. Frame-level approaches classify each audio frame/segment into event classes and then aggregate the consecutive frame-level predictions to identify sound event boundaries or endpoints (Miyazaki et al. 2020a; Lin et al. 2019). They are often heavily manually designed with plenty of heuristics and data-specific parameter optimization, hence less scalable and reliable across different audio data. Event-level approaches, on the other hand, directly model the temporal boundaries of sound events, taking into

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

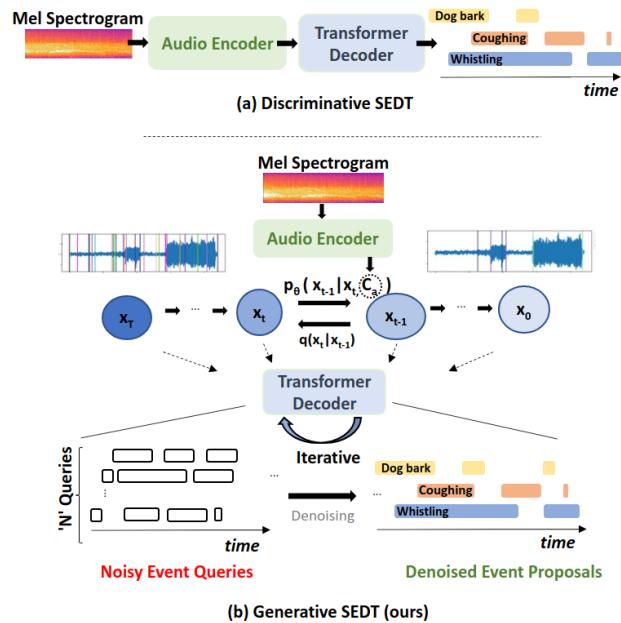


Figure 1: Architectural comparison: (a) Conventional discriminative DETR-based Sound Event Detector Transformer (SEDT) (Ye et al. 2021) incorporates a single decoding step with clean queries. (b) Our diffusion-infused generative DETR-based Sound Event Detector (DiffSED) conducts multi-step decoding/denoising over noised queries.

account the correlation between frames, thereby eliminating the mundane post-processing step and are more generalizable (Ye et al. 2021). In both approaches, existing methods rely on *proposal prediction* by regressing the start and end times of each, *i.e.*, discriminative learning based.

Recently, generative learning models such as diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020) have emerged strongly in computer vision. Conceptually, we draw an analogy between the SED problem and image-based object detection (Duan et al. 2019; Chen et al. 2019). We consider the latest generative learning based object detection approach (Chen et al. 2022b) represents a new direction for designing detection models in general. Although conceptually similar to object detection, the SED

problem still presents unique challenges and complexity due to the presence of temporal dynamics. Besides, there are several limitations with the detection diffusion formulation in (Chen et al. 2022b). First, a two-stage pipeline (*e.g.*, RCNN (Chao et al. 2018)) is adopted, giving rise to localization-error propagation from proposal generation to proposal classification (Nag et al. 2022). Second, as each event proposal is processed individually, their intrinsic relationship modeling is overlooked, potentially hurting the learning efficacy. To address these issues, we present two different designs: (a) Adopting the one-stage detection pipeline (Tian et al. 2019; Wang et al. 2020) that have already shown excellent performance with a relatively simpler design, in particular, DETR (Carion et al. 2020). Even within the SED literature, this simpler pipeline has shown to achieve higher accuracy than frame-level models on a variety of sound event detection datasets due to the better temporal resolution, as well as its ability to learn long-range dependencies between sound events (Ye et al. 2021). (b) A unique challenge with SED is **big boundary ambiguity** as compared to object detection. This is because temporal audio events are continuous in time without clear start and end points (*e.g.*, non-zero momentum), and the transition between consecutive events is often stochastic. Further, human perception of event boundaries is also instinctive and subjective. For the above reasons, we reckon that diffusion-based models could be a great fit for sound event detection.

Nonetheless, it is non-trivial to integrate denoising diffusion with existing sound event detection models, due to several reasons. (1) Whilst efficient at processing high-dimension data simultaneously, diffusion models (Dhariwal and Nichol 2021; Li et al. 2022) have typically been shown to work with continuous input data. But event boundaries in SED are discrete. (2) Denoising diffusion and SED both suffer low efficiency, and their combination would even get worse. Both of the problems have not been investigated systematically thus far.

To address the aforementioned challenges, a novel *conditioned event diffusion* method is proposed for efficiently tackling the SED task, abbreviated as **DiffSED**. In the forward diffusion process, Gaussian noises are added to the event latents iteratively. In the reverse denoising process, the noisy latents are passed as queries to a denoiser (*e.g.*, DETR (Carion et al. 2020)) for denoising the event latents so that desired event proposals can be obtained, with the condition on the observation of an input audio stream. The usage of noisy latents allows our model to bypass the need for continuous input, as the denoising diffusion process takes place in the designated latent space. During inference, the model can take as input the noisy latents composed of noises sampled from Gaussian distribution and learned components, and outputs the event proposals of a given audio stream (*i.e.*, the condition). The proposed noise-to-queries strategy for denoising diffusion has several appealing properties: (i) Evolutionary enhancement of queries during inference wherein each denoising step can be interpreted as a unique distribution of noise thus adding stochasticity to solve the boundary ambiguity problem. (ii) Integrating denoising diffusion with this noisy-latent decoder design solves the typical slow-

convergence limitation.

We summarize the **contributions** of this work. **(a)** We reformulate sound event detection (SED) as a generative denoising process (see Fig. 1) in an elegant transformer decoder framework. This is the first study to apply the diffusion model for the SED task to the best of our knowledge. **(b)** The proposed generative adaptation uses a noise-to-queries strategy with several appealing properties such as evolutionary enhancement of queries and faster convergence. **(c)** Our comprehensive experiments on the URBAN-SED (Salamon, Jacoby, and Bello 2014) and the EPIC-Sounds (Huh et al. 2023) datasets validate the significant performance advantage of our DiffSED over existing alternatives.

Related Work

Sound Event Detection

The existing SED literature can be divided into two categories, namely, frame-level approaches and event-level approaches. In frame-level approaches (Lim, Park, and Han 2017; Turpault et al. 2019; Miyazaki et al. 2020a), the input audio signal is first divided into short, fixed-length segments, and the sound events within each segment are further classified *independently*. Despite strong performance and good intuition, this split-and-classify strategy requires plenty of heuristics designs, unscalable parameter settings (*e.g.*, segment duration), as well as time-consuming post-processing (*e.g.*, aggregating frame-level predictions). To overcome these limitations, event-level approaches (Ye et al. 2021) present a more principled and scalable solution with end-to-end learning frameworks, inspired by the model designs in object detection (Carion et al. 2020; Zhu et al. 2020; Zhang et al. 2022) and video action recognition domains (Tan et al. 2021; Shi et al. 2022). Whilst being understudied, this strategy has shown to be more efficient and robust to longer and more complex (overlapping) events, such as those in music and human speech as well as short and frequently occurring events such as those in urban soundscapes or environmental monitoring. Our DiffSED belongs to this category, further pushing this forefront of performance.

Deep learning techniques have achieved excellent performance in SED. For instance, convolutional neural networks (CNNs) have been widely investigated for audio event classification (Cakir et al. 2017; Kumar, Khadkevich, and Fügen 2018) owing to their ability to efficiently capture and analyze local patterns within the acoustic waveform of sound. Additionally, recurrent neural networks (RNNs) have been used for temporal modeling of audio signals in arrears to their propensity to capture long-term temporal dependencies in sequential data - an innate property of audio signals. Interestingly, apart from the hybrid approaches (Li et al. 2020; Koh et al. 2021), that utilize CNNs to extract features from the audio signal, which are then fed into an RNN to model temporal dependencies, recently, transformer based architectures (Wakayama and Saito 2022; Chen et al. 2022a) have been shown as equally promising, particularly, leveraging the self-attention mechanisms to model temporal relationships in audio signals and capturing complex patterns over time. Commonly, all the prior methods consider the SED

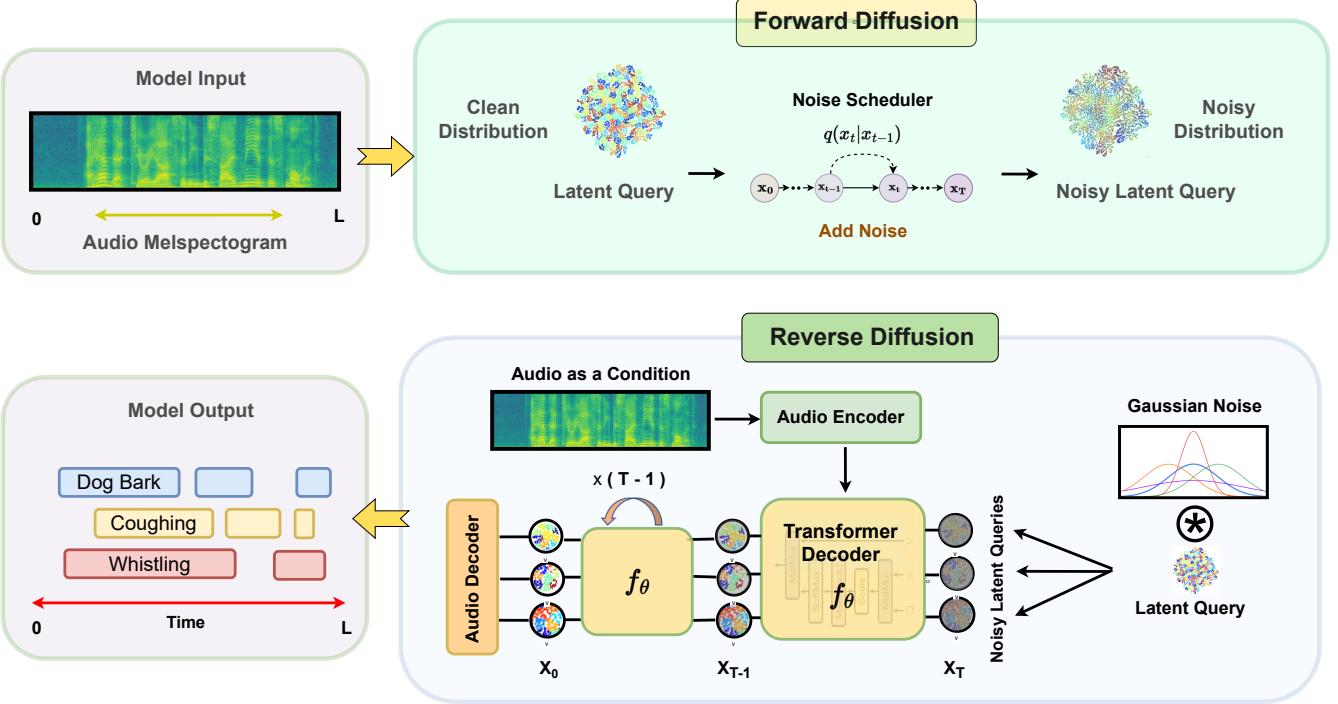


Figure 2: Overview of our proposed DiffSED. (Top) In the forward diffusion process, Gaussian noises are added to the event latents iteratively to obtain noisy latents X_T . (Bottom) In the reverse denoising process, an audio melspectrogram is passed as the condition along with random noisy latents sampled from the Gaussian distribution. The noisy latents are passed as the query to the denoiser for denoising the event latents in an iterative fashion to obtain event proposals.

problem as discriminative learning. In contrast, we treat for the first time this problem in a unique perspective of generative learning. In particular, we generate the sound event bounds and predict the class labels from noise latents, with the condition to the input audio sample.

Diffusion-Based Models for Audio Tasks

As a new class of deep generative models, diffusion models have been gaining popularity in different fields. Beginning with a sample from a random distribution, the diffusion model is optimized to gradually learn a denoising schedule to obtain a noise-free target. This paradigm has yielded remarkable results in audio processing tasks ranging from audio generation (Leng et al. 2022; Huang et al. 2022), audio enhancement (Lemercier et al. 2022), audio separation (Lutati, Nachmani, and Wolf 2023) etc. To the best of our knowledge, this is the first work that exploits a diffusion model for the SED task.

Methodology

Problem Definition Sound event detection (SED) involves both classification and *temporal* localization given an audio sequence. In this task, the audio sequence is usually represented as a 2-dimensional feature, such as a mel-spectrogram. We want a model to output the onset and offset times of all target events and the corresponding event labels (Wakayama and Saito 2022). To train the model, we collect

a set of labeled audio sequence set $D^{train} = \{A_i, \psi_i\}$. Each audio $A_i \in \mathcal{R}^{T \times F}$ (where $T \times F$ represents the spectro-temporal dimension) is labeled with temporal annotation $\psi_i = \{(\Psi_j, \xi_j, y_j)\}_{j=1}^{M_i}$ where Ψ_j / ξ_j represents onset/off-set of an event and y_j denotes the acoustic class event label.

Preliminaries on Diffusion Model

Diffusion models are a class of generative models that use the diffusion process to model complex probability distributions (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020). In a diffusion model, the forward process generates samples by iteratively applying a diffusion equation to a starting noise vector. The forward process can be represented by the following equation:

$$z_t = \sqrt{1 - \beta_t} * z_{t-1} + \sqrt{\beta_t} * x_t \quad (1)$$

where z_t is the diffusion state at time t , x_t is the input at time t , and β_t is the diffusion coefficient at time t . The noise scale is controlled by β_t which adopts a monotonically decreasing cosine schedule (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020) in every different time step t .

Denoising in diffusion models is the process of generating a clean representation from a noisy observation by reversing the diffusion process. In other words, the goal is to obtain an estimate of the original representation from the final diffusion state. The denoising process can be performed using the reverse diffusion process, which can be represented by

Algorithm 1: Training

```

1  def train_loss(audio, event_query):
2  """
3  audio: [B, T, F]
4  event_queries: [B, N, D]
5  # B: batch_size
6  # N: number of event queries
7  """
8  # Encode audio features
9  audio_feats = audio_encoder(audio)
10 # Signal scaling
11 event_queries = (event_queries * 2 - 1) *
12   scale
13 # Corrupt event_queries
14 t = randint(0, T)           # time step
15 eps = normal(mean=0, std=1) # noise: [B, N, D]
16 event_queries_crpt = sqrt(    alpha_cumprod(t)
17   ) * event_queries +
18   sqrt(1 - alpha_cumprod(t)) * eps
19 # Predict bounding boxes
20 pb_pred = detection_decoder(
21   event_queries_crpt, audio_feats, t)
22 # Set prediction loss
23 loss = set_prediction_loss(pb_pred, gt_boxes)
24 return loss

```

the following equation:

$$z_{T-t} = (z_{T-t+1} - \sqrt{\beta_{T-t}} * x_{T-t}) / \sqrt{1 - \beta_{T-t}} \quad (2)$$

where z_T is the final diffusion state, x_t is the noisy input at time t , and β_{T-t} is the diffusion coefficient at time $T-t$. The denoising process starts from the final diffusion state z_T and iteratively applies the reverse diffusion equation to obtain an estimate of the original representation $x_0 = z_1$.

$$x_t = (z_{t+1} - \sqrt{\beta_t} * x_{t-1}) / \sqrt{1 - \beta_t} \quad (3)$$

where $\forall t \in [1, T-1]$ such that x_t is the estimate of the original representation at time t . The denoising process can be improved by adding regularization or constraints to the estimate of the original representation.

DiffSED: Architecture Design

Diffusion-Based SED Formulation In this work, we formulate the SED task in a conditional denoising diffusion framework. In our setting, data samples are a set of learnable event query embeddings $\mathbf{z}_0 = b$, where $b \in \mathbb{R}^{N \times D}$ denotes N event query embeddings at the dimension of D . In our implementation, the event queries are retrieved from a simple lookup table that stores embeddings of a fixed dictionary of size N (initialized from $\mathcal{N}(0, 1)$). A neural network $f_\theta(\mathbf{z}_t, t, A)$ is trained to predict \mathbf{z}_0 from noisy proposals \mathbf{z}_t , conditioned on the corresponding audio A . The audio category \hat{y} is predicted subsequently. See Algorithm 1 for more details.

Since the diffusion model generates a data sample iteratively, it needs to run the model f_θ multiple times in inference. It would be computationally intractable to directly apply f_θ on the raw audio at every iterative step. For efficiency, we propose to separate the whole model into two parts, *audio encoder* and *detection decoder*, where the former runs only once to extract a feature representation of the

input audio A_i , and the latter takes this feature as a condition to progressively refine the noisy proposals \mathbf{z}_t (please refer Fig. 2).

Audio Encoder The audio encoder takes as input the pre-extracted audio mel-spectograms and extracts high-level features for the following detection decoder. In general, any audio encoder can be used. We follow (Ye et al. 2021) for the audio encoder. More specifically, the raw audio is first encoded using a CNN based encoder backbone (*i.e.*, ResNet-50) to obtain the audio feature $A_f \in \mathbb{R}^{T' \times F'}$ respectively. This is followed by a multi-layered temporal transformer (Vaswani et al. 2017) τ that performs global attention across the time dimension to obtain the global feature as:

$$C_a = \tau(A_f) \quad (4)$$

where query, key, and value of the transformer is set to A_f . We also append positional encoding to A_f before passing it into the transformer.

Detection Decoder Similar to SEDT (Ye et al. 2021), we use a transformer decoder (Vaswani et al. 2017) (denoted by f_θ) for detection. Functionally, in our formulation it serves as a denoiser. In traditional DETR (Lin et al. 2021), the queries are learnable continuous embeddings with random initialization. In DiffSED, however, we exploit the queries as the *denoising targets*.

As opposed to adding noises to object boundaries (Chen et al. 2022b), we inject the Gaussian noise to the randomly initialized latent queries. This is similar to the concept of *event queries* (Rombach et al. 2022). To detect multiple events occurring simultaneously, we sample N such *noisy event queries* to form $Q \in \mathbb{R}^{N \times D}$ which will be subsequently passed on to the detection decoder for denoising. Taking Q as input, the decoder predicts N outputs:

$$F_d = f_\theta(Q; C_a) \in \mathbb{R}^{N \times D} \quad (5)$$

where C_a is the encoded audio feature and the F_d is the final embedding. F_d is finally decoded using two parallel heads namely (1) event classification head and (2) event localization head respectively. The first estimates the probability of a particular event within the event proposal. The second estimates the onset and offset of event in the raw audio.

Model Training

During training, we first construct the diffusion process that corrupts the event latents to noisy latents. We then train the model to reverse this noising process. We add Gaussian noises to the learnable queries. The noise scale is controlled by β_t (Eq. (1)), which adopts a monotonically decreasing cosine schedule in different timestep t , following (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020). The decoder uses the noisy event queries (corresponding to t) and the global feature C_a as the condition (see Fig 1 (b)) to generate the denoised event queries (corresponding to $t-1$) repeatedly until an approximation of Q is obtained. The output from the last denoising step (corresponding to each input event query) is projected into sigmoidal onset and offset timestamps and an event probability distribution using separate feedforward projection layers. We observe that SED

favors a relatively high signal scaling value than object detection (Chen et al. 2022b) (see Table 5).

The event-based objective is defined as a combination of a binary classification loss for event onset and offset prediction and a cross-entropy loss for event class prediction. We compute Hungarian assignment between ground truth boxes and the outputs of the model. We supervise the model training using each pair of matched ground-truth/prediction (event class and the temporal boundary).

Model Inference

In inference, the noisy event queries are randomly sampled from a Gaussian distribution. Starting from noisy latents sampled from a Gaussian distribution, the model progressively refines the predictions. At each sampling step, the random or estimated latents from the last sampling step are sent into the detection decoder to predict the event category and the event onset/offsets. After obtaining the event proposals of the current step, DDIM (Song, Meng, and Ermon 2021) is adopted to estimate the proposals for the next step. DiffSED has a simple event proposal generation pipeline without post-processing (e.g., non-maximum suppression).

Key Insights

One Model Multiple Trade-Offs Once trained, DiffSED works under a varying number of event queries and sampling steps in inference. While inferring, each sampling step involves, estimating event queries from the last sampling step and sending them back into the detection decoder to eventually predict the event classes and event boundaries at the t_0 step, *i.e.*, fully denoised. In general, better accuracy can be obtained using more queries and fewer steps (see Table 3 and Table 4). We discuss the multistep decoding experiments in detail in our ablation study. Ultimately, it can be determined that a single DiffSED can meet a number of different trade-off needs between speed and accuracy.

Faster Convergence DETR-style detection models suffer generally slow convergence (Liu et al. 2022) due to inconsistent matching of event queries to the event proposals. Concretely, for the same audio, an event query is often matched with different event boundaries in different epochs, making the optimization oscillating and difficult. In DiffSED each query is designed as a proposal proxy – a noised event query that can be regarded as a good event proposal due to staying close to the corresponding ground truth boundary. Our query denoising task thus has a definite optimization objective which is the ground truth proposal. We validate that query denoising based DiffSED converges faster than SEDT (see Fig 3), whilst achieving superior performance (Table 1).

Experiments

Datasets We present our results on two datasets namely, URBAN-SED (Salamon, Jacoby, and Bello 2014) and EPIC-Sounds (Huh et al. 2023). *URBAN-SED* is a publicly available dataset for SED in urban environments. It is accompanied by detailed annotations, including onset and offset times for each sound event, along with human generated accurate annotations. The *EPIC-Sounds* dataset consists of

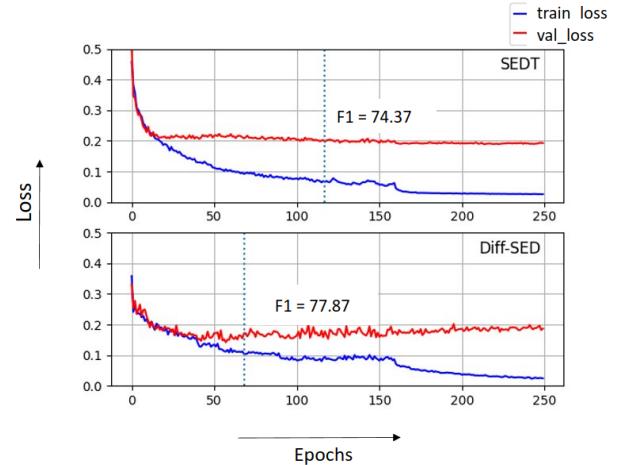


Figure 3: Convergence rates for SEDT and DiffSED on the URBAN-SED dataset. The dotted lines represent the training epoch when the best-performing checkpoint (the one with the best audio-tagging F1 score on the validation set) arrived. DiffSED trains faster (>40%) and achieves better optimum than SEDT.

more than 36,000 audio recordings of various lengths, totaling over 500 hours of audio. The recordings were made in a variety of indoor and outdoor environments, including office spaces, public places, and natural environments. They cover a wide range of sound classes, including human speech, animal sounds, environmental sounds, and music.

Evaluation Metrics To evaluate the model’s performance on the URBAN-SED dataset, we measure F1-score, precision, and recall for both event-level and segment-level settings on the test split. For the EPIC-Sounds dataset, we report the top-1 and top-5 accuracy, as well as mean average precision (mAP), mean area under ROC curve (mAUC), and mean per class accuracy (mCA) on the validation split, following the protocol of (Huh et al. 2023).

Implementation Details

Training Schedule We use a pre-trained encoder backbone ResNet-50 for feature extraction, for fair comparisons with previous methods (Ye et al. 2021). Our model is trained for 400 epochs, while re-initializing the weights from the best checkpoint for every 100 epochs, using Adam optimizer with an initial learning rate of 10^{-4} with a decay schedule of 10^{-2} . The batch size is set to 64 for URBAN-SED and 128 for EPIC-Sounds. All models are trained with 2 NVIDIA-A5500 GPUs.

Testing Schedule At the inference stage, the detection decoder iteratively refines the predictions from Gaussian random latent queries. For efficiency, by default, we denoise for a single time-step, *i.e.*, $T_0 \leftarrow T_{1000}$ timestep.

Model	Event-based [%]			Segment-based [%]			Audio tagging[%] F1
	F1	P	R	F1	P	R	
CRNN-CWin (Miyazaki et al. 2020b)	36.75	—	—	65.74	—	—	74.19
Ctrans-CWin (Miyazaki et al. 2020b)	34.36	—	—	64.73	—	—	74.05
SEDT (Ye et al. 2021)	37.27	43.32	33.21	65.21	74.82	58.46	74.37
DiffSED (Ours)	43.89	48.46	37.82	69.24	77.49	62.05	77.87

Table 1: Results on URBAN-SED (Test set)

Model	Top-1	Top-5	mCA	mAP	mAUC
ASF (Kazakos et al. 2021)	53.47	84.56	20.22	0.235	0.879
SSAST (Gong et al. 2022)	53.75	84.54	20.11	0.254	0.873
DiffSED (Ours)	56.85	87.45	20.75	0.277	0.861

Table 2: Results on EPIC-Sounds (Validation set)

Algorithm 2: Noise corruption

```

1  \resizebox{0.75\linewidth}{!}{!
2  def add_noise():
3      """
4      gt_boxes: [B, *, 2]
5      event_queries: [B, N, D]
6      B: batch_size
7      N: number of event queries
8      """
9      if corrupt bounding_boxes: # Diff-SED-BB
10         # Padding (repeat) bounding boxes
11         pb = Pad(gt_boxes, N) #[B, N, 2]
12         # Signal scaling
13         pb = (pb * 2 - 1) * scale
14         # Corrupt bounding boxes
15         t = randint(0, T)          #time step
16         eps = normal(mean=0, std=1) #noise: [B, N, 2]
17         pb_crpt = sqrt(alpha_cumprod(t)) * pb + sqrt(1
18             - alpha_cumprod(t)) * eps
19         event_queries_crpt = Project(pb_crpt)
20         #[B, N, 2] -> [B, N, D]
21     else: # DiffSED
22         # Signal scaling
23         event_queries = (event_queries * 2 - 1) *
24             scale
25         # Corrupt event_queries
26         t = randint(0, T)          #time step
27         eps = normal(mean=0, std=1) #noise: [B, N, D]
28         event_queries_crpt = sqrt(alpha_cumprod(t)) *
29             event_queries +
30             sqrt(1 - alpha_cumprod(t)) * eps
31     return event_queries_crpt

```

Main Results

Results on URBAN-SED We compare our model with previous end-to-end approaches under the supervised learning setting. The primary contribution of our work lies in proposing a diffusion-infused transformer decoder that provides a more robust representation of grounded event boundaries in the encoded acoustic features. From Table 1, we draw the following conclusions: (1) The diffusion-based decoder of DiffSED performs significantly better than all the other methods for both event-level and segment-level met-

rics, with a 6.62% and 4.03% absolute improvement, respectively. (2) Additionally, our model outperforms existing approaches in terms of audio-tagging results, with a 3.5% absolute improvement. This validates our model formulation in exploiting the SED problem as generative learning in the denoising diffusion framework.

Results on EPIC-Sounds We use the publicly available pre-trained backbones ASF (Kazakos et al. 2021) and SSAST (Gong et al. 2022) as competing models. We observe from Table 2 that: (1) DiffSED consistently outperforms both the alternatives with 3.1% and 2.89% improvement in the Top-1 and Top-5 accuracies, respectively; (2) Our model performs competitively in the mAUC score.

Ablation Study

We conduct ablation experiments on URBAN-SED to study DiffSED in detail. All experiments use the pre-trained ResNet-50 backbone features for training and inference without further specification.

Denoising Strategy Due to the inherent query based design with the detection decoder, we discuss and compare two denoising strategies: (1) Corrupting the event latents in the continuous space and passing it as queries (referred as DiffSED, our choice). (2) Corrupting discrete event proposals (*i.e.*, ground-truth bounding boxes) and projecting it as queries (denoted as DiffSED-BB, detailed in Algorithm 2). Additionally, we corrupt the label queries using random shuffle as the noise in the forward diffusion step. To evaluate the effect of the denoising strategy experimentally, we test both variants using different numbers of event proposals. It can be observed in Table 3 that both variants achieve the best audio-tagging performance when using 30 event proposals as input to the decoder. Also, the overall scores in both event-level and segment-level metrics are lesser for DiffSED-BB compared to DiffSED. We hypothesize this is caused by some adversarial effect in projecting the ground-truth bounding box (2-dimensional) to the latent event query.

	#Queries	Event-F1[%]	Segment-F1[%]	AT[%]
DiffSED-BB	10	31.43	58.85	68.87
	20	35.32	60.53	68.84
	30	37.29	60.91	69.61
	40	31.95	58.79	68.41
	50	31.81	57.89	68.31
DiffSED	10	40.78	68.41	77.22
	20	41.42	68.73	76.54
	30	41.3	68.21	77.46
	40	38.65	67.21	75.21
	50	36.28	64.22	72.77

Table 3: Effect of the number of queries on the performance for URBAN-SED Test set. (AT: Audio Tagging performance)

Multistep Decoding We tabulate the results upon varying the number of denoising steps for both DiffSED and DiffSED-BB in Table 4. We observe a steady improvement over the event-level and segment-level F1 scores as we increase the number of denoising steps from 1 to 5 and then gradually decrease when using 10 decoding steps. However, the best audio tagging performance is achieved when performing a single-step decoding. We hypothesize this is primarily because the event boundaries have short-range temporal dependencies that might not benefit significantly from multistep denoising. The noise addition mainly affects each time step independently and doesn't accumulate over multiple steps hence does not yield substantial improvements. Denoising over multiple timesteps requires more computing, while providing only a marginal gain thus not worthwhile.

	#steps	Event-F1[%]	Segment-F1[%]	AT[%]
DiffSED-BB	1	39.78	64.74	72.92
	5	38.27	65.72	71.88
	10	38.3	64.82	72.17
DiffSED	1	43.89	69.24	77.87
	5	44.35	70.75	77.07
	10	43.50	69.05	77.36

Table 4: Effect of the number of denoising steps used while inference on the performance for URBAN-SED Test set. (AT: Audio Tagging performance)

Signal Scaling The signal scaling factor controls the signal-to-noise ratio (SNR) of the diffusion process. We study the influence of scaling factors. The results in Table 5 demonstrate that the scaling factor of 0.4 achieves the highest audio-tagging performance as well as all other metrics for DiffSED, whereas for DiffSED-BB the best audio tagging performance is obtained for a scaling factor of 0.2 whilst achieving the best event-level and segment-level F1 score for a scaling factor of 0.4. This suggests the relationship between optimal scaling and the denoising strategy.

	Noise scale	Event-F1[%]	Segment-F1[%]	AT[%]
DiffSED-BB	0.1	32.61	32.45	73.49
	0.2	35.91	35.73	75.73
	0.3	37.29	60.91	69.61
	0.4	39.78	64.74	72.92
	0.5	33.14	61.79	71.12
DiffSED	0.1	37.61	54.63	72.2
	0.2	39.65	58.17	73.89
	0.3	41.3	68.21	77.46
	0.4	43.89	69.22	77.87
	0.5	39.23	59.25	72.78

Table 5: Effect of scaling the noise factor on the performance for URBAN-SED Test set. (AT: Audio Tagging Performance)

	Runs	Event-F1[%]	Segment-F1[%]	AT[%]
DiffSED-BB	1	38.6($\uparrow 0.2$)	64.32($\downarrow 0.09$)	72.48(0.0)
	2	39.45($\downarrow 0.57$)	64.15($\uparrow 0.07$)	72.88($\downarrow 0.4$)
	3	38.57($\uparrow 0.3$)	64.21($\uparrow 0.01$)	72.08($\uparrow 0.4$)
DiffSED	Avg	38.87	64.22	72.48
	1	43.12($\downarrow 0.2$)	68.38($\uparrow 0.5$)	77.62($\downarrow 0.01$)
	2	42.35($\uparrow 0.5$)	68.97($\uparrow 0.01$)	77.59($\uparrow 0.02$)
	3	43.29($\downarrow 0.3$)	69.54($\downarrow 0.5$)	77.62($\downarrow 0.01$)
	Avg	42.92	68.96	77.61

Table 6: Effect of changing the seed value for inducing noise during inference. Values inside (.) indicate deviation from the mean calculated over 3 runs.

Random Seed DiffSED starts with random noisy event queries as input during inference. We evaluate the stability of DiffSED and DiffSED-BB by training three models independently with strictly the same configurations (30 noisy event proposals as input to the decoder and a scaling factor of 0.4) except for random seed on URBAN-SED dataset. Then, we evaluate each model instance with 3 different random seeds to measure the distribution of performance, inspired by (Chen et al. 2022b). As shown in Table 6, most evaluation results are distributed closely to the average metrics for both variants. This demonstrates that our models are robust to random event queries.

Conclusion

In this work, we reformulate the Sound Event Detection (SED) problem from the generative learning perspective, in particular under the diffusion-based transformer framework. We introduce a diffusion adaptation method characterized by noisy event latents denoising. This design has the advantage of being able to model the global dependencies of sound events, while still being computationally efficient. Our study verifies the efficacy of diffusion models in a new problem context (*i.e.*, SED), consistent with previous findings. Experiments show that our method is superior to existing art alternatives on standard benchmarks.

References

Bear, H. L.; Nolasco, I.; and Benetos, E. 2019. Towards joint sound scene and polyphonic sound event recognition. In *INTERSPEECH*.

Bhosale, S.; Chakraborty, R.; and Kopparapu, S. K. 2023. A Novel Metric For Evaluating Audio Caption Similarity. In *IEEE ICASSP*.

Cakir, E.; Parascandolo, G.; Heittola, T.; Huttunen, H.; and Virtanen, T. 2017. Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the faster r-cnn architecture for temporal action localization. In *IEEE CVPR*.

Chen, K.; Du, X.; Zhu, B.; Ma, Z.; Berg-Kirkpatrick, T.; and Dubnov, S. 2022a. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *IEEE ICASSP*.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C. C.; and Lin, D. 2019. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155*.

Chen, S.; Sun, P.; Song, Y.; and Luo, P. 2022b. Diffusion-det: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *NeurIPS*.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6569–6578.

Gong, Y.; Lai, C.-I.; Chung, Y.-A.; and Glass, J. 2022. Ssast: Self-supervised audio spectrogram transformer. In *AAAI Conference on Artificial Intelligence*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*.

Huang, R.; Lam, M. W.; Wang, J.; Su, D.; Yu, D.; Ren, Y.; and Zhao, Z. 2022. Fastdiff: A fast conditional diffusion model for high-quality speech synthesis. *International Joint Conferences on Artificial Intelligence (IJCAI)*.

Huh, J.; Chalk, J.; Kazakos, E.; Damen, D.; and Zisserman, A. 2023. EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound. In *IEEE ICASSP*.

Igarashi, A.; Imoto, K.; Komatsu, Y.; Tsubaki, S.; Hario, S.; and Komatsu, T. 2022. How Information on Acoustic Scenes and Sound Events Mutually Benefits Event Detection and Scene Classification Tasks. In *IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

Kazakos, E.; Nagrani, A.; Zisserman, A.; and Damen, D. 2021. Slow-fast auditory streams for audio recognition. In *IEEE ICASSP*.

Koh, C.-Y.; Chen, Y.-S.; Liu, Y.-W.; and Bai, M. R. 2021. Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks. In *IEEE ICASSP*.

Kumar, A.; Khadkevich, M.; and Fügen, C. 2018. Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes. In *IEEE ICASSP*.

Lemercier, J.-M.; Richter, J.; Welker, S.; and Gerkmann, T. 2022. Analysing Diffusion-based Generative Approaches versus Discriminative Approaches for Speech Restoration. In *IEEE ICASSP*.

Leng, Y.; Chen, Z.; Guo, J.; Liu, H.; Chen, J.; Tan, X.; Mandic, D.; He, L.; Li, X.-Y.; Qin, T.; et al. 2022. Binauralgrad: A two-stage conditional diffusion probabilistic model for binaural audio synthesis. *Neural Information Processing Systems (NeurIPS)*.

Li, X. L.; Thickstun, J.; Gulrajani, I.; Liang, P.; and Hashimoto, T. B. 2022. Diffusion-LM Improves Controllable Text Generation. *arXiv preprint arXiv:2205.14217*.

Li, Y.; Liu, M.; Drossos, K.; and Virtanen, T. 2020. Sound event detection via dilated convolutional recurrent neural networks. In *IEEE ICASSP*.

Lim, H.; Park, J.-S.; and Han, Y. 2017. Rare Sound Event Detection Using 1D Convolutional Recurrent Neural Networks. In *Detection Classification Acoustic Scenes Events (DCASE) Workshop*.

Lin, L.; Wang, X.; Liu, H.; and Qian, Y. 2019. Guided learning convolution system for dcase 2019 task 4. *arXiv preprint arXiv:1909.06178*.

Lin, M.; Li, C.; Bu, X.; Sun, M.; Lin, C.; Yan, J.; Ouyang, W.; and Deng, Z. 2021. DETR for Crowd Pedestrian Detection. *arXiv:2012.06785*.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In *International Conference on Learning Representations*.

Lutati, S.; Nachmani, E.; and Wolf, L. 2023. Separate And Diffuse: Using a Pretrained Diffusion Model for Improving Source Separation. *arXiv preprint arXiv:2301.10752*.

Mesaros, A.; Heittola, T.; Virtanen, T.; and Plumbley, M. D. 2021. Sound event detection: A tutorial. *IEEE Signal Processing Magazine*.

Miyazaki, K.; Komatsu, T.; Hayashi, T.; Watanabe, S.; Toda, T.; and Takeda, K. 2020a. Convolution-augmented transformer for semi-supervised sound event detection. In *Detection Classification Acoustic Scenes Events (DCASE) Workshop*.

Miyazaki, K.; Komatsu, T.; Hayashi, T.; Watanabe, S.; Toda, T.; and Takeda, K. 2020b. Weakly-supervised sound event detection with self-attention. In *IEEE ICASSP*.

Nag, S.; Zhu, X.; Song, Y.-Z.; and Xiang, T. 2022. Proposal-free temporal action detection via global segmentation mask learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, 645–662. Springer.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Salamon, J.; Jacoby, C.; and Bello, J. P. 2014. A dataset and taxonomy for urban sound research. In *ACM International Conference on Multimedia*.

Shi, D.; Zhong, Y.; Cao, Q.; Zhang, J.; Ma, L.; Li, J.; and Tao, D. 2022. ReAct: Temporal Action Detection with Relational Queries. In *European conference on computer vision*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Tan, J.; Tang, J.; Wang, L.; and Wu, G. 2021. Relaxed transformer decoders for direct action proposal generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636.

Turpault, N.; Serizel, R.; Salamon, J.; and Shah, A. P. 2019. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. *Detection Classification Acoustic Scenes Events (DCASE) Workshop*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wakayama, K.; and Saito, S. 2022. CNN-Transformer with Self-Attention Network for Sound Event Detection. In *IEEE ICASSP*.

Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; and Li, L. 2020. Solo: Segmenting objects by locations. In *ECCV*.

Xie, Z.; Xu, X.; Wu, M.; and Yu, K. 2023. Enhance Temporal Relations in Audio Captioning with Sound Event Detection. *arXiv preprint arXiv:2306.01533*.

Xu, X.; Dinkel, H.; Wu, M.; and Yu, K. 2021. Text-to-audio grounding: Building correspondence between captions and sound events. In *IEEE ICASSP*.

Ye, Z.; Wang, X.; Liu, H.; Qian, Y.; Tao, R.; Yan, L.; and Ouchi, K. 2021. Sound Event Detection Transformer: An Event-based End-to-End Model for Sound Event Detection. *arXiv preprint arXiv:2110.02011*.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.; and Shum, H. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv preprint*.