Diffusion-Based Particle-DETR for BEV Perception

Asen Nachkov[1] Danda Pani Paudel[1] Martin Danelljan[2] Luc Van Gool[1,2]

[1]INSAIT, Sofia University [2]ETH Zurich

Abstract

The Bird-Eye-View (BEV) is one of the most widely-used scene representations for visual perception in Autonomous Vehicles (AVs) due to its well suited compatibility to downstream tasks. For the enhanced safety of AVs, modeling perception uncertainty in BEV is crucial. Recent diffusion-based methods offer a promising approach to uncertainty modeling for visual perception but fail to effectively detect small objects in the large coverage of the BEV. Such degradation of performance can be attributed primarily to the specific network architectures and the matching strategy used when training. Here, we address this problem by combining the diffusion paradigm with current state-of-the-art 3D object detectors in BEV. We analyze the unique challenges of this approach, which do not exist with deterministic detectors, and present a simple technique based on object query interpolation that allows the model to learn positional dependencies even in the presence of the diffusion noise. Based on this, we present a diffusion-based DETR model for object detection that bears similarities to particle methods. Abundant experimentation on the NuScenes dataset shows equal or better performance for our generative approach, compared to deterministic state-of-the-art methods. The source code is at https://github.com/insait-institute/ParticleDETR.

1. Introduction

3D Object detection - the task of localizing and classifying objects in a real-world 3D coordinate frame - is one of the most important tasks in the pipeline of an autonomous vehicle. It is critical to safe self-driving since it informs the subsequent prediction, planning, and actuation modules and, evidently, one needs to recognize an obstacle in order to avoid it. Estimating the object locations directly from the camera views [2, 3, 43] faces difficulties related to perspective warping and size-distance ambiguities. Instead, the bird-eye-view (BEV) has established itself as a useful representation to facilitate perception because it is ego-centered, metrically-accurate, orthographic - thus avoiding perspective distortion of shapes, and suffers less from occlusions and object deformations.

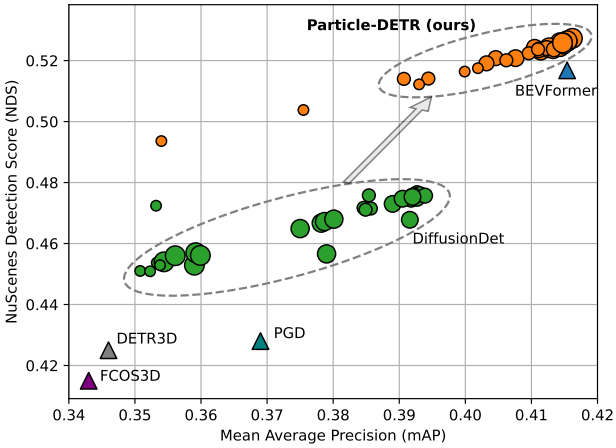

Figure 1. **Performance improvement**. Our method outperforms the baseline stochastic model DiffusionDet [9] and is comparable in performance to deterministic models like BEVFormer [34]. Here, the results of stochastic models are shown as circles and those of deterministic models as triangles. The size of the circles is proportional to the number of search tokens used.

Recently, it has been shown that diffusion models can be successfully used for 2D object detection [9] - a completely different setup than the generative tasks like text-to-image where they have been dominating [11, 25, 53, 54, 56]. In principle, one should then be able to apply diffusion-based object detection also in the 2D BEV and predict 3D locations, reaping all the diffusion benefits like incremental refinement and the ability to trade-off compute and accuracy?

We find that naively applying diffusion in BEV yields an algorithm with insufficient performance. We attribute this to the challenging problem setting and the fact that the network architecture is not tuned for the particular geometric aspects of the BEV:

1. **Setting**: Recent works [33, 34, 48, 52] represent the BEV as a set of spatially-correlated latent features corresponding to a (50×50) or even (100×100) meter grid around the ego-vehicle. The detectable objects such as cars and pedestrians are naturally very small in relation to the size of the whole BEV map, mak-

ing detection more challenging compared to on common datasets used to benchmark 2D detection algorithms [16, 17, 37].

2. **Architecture**: DiffusionDet [9], the representative algorithm, uses an ROI-based architecture [21], aggregating BEV features only within the proposed boxes. This makes object features local, preventing extensive search on the BEV. Local features work well in settings where the target boxes are larger and more dense, but we believe in the BEV one needs a more specialized architecture to better handle object sparsity.

Problem statement and approach. Since object detection is ultimately a search problem and smaller objects are harder to locate, some of the inherent challenges when using diffusion to detect objects can be exacerbated in the BEV. Thus, the research question we try to answer is: *How should the diffusion approach and network architecture be adjusted so as to ease the search process in the BEV?* To that end, our insights are that first, to search more effectively, one should pool information across the *search tokens* used (boxes, anchors, queries), and second, one should take measures to prevent the diffusion noise from overwhelming any positional dependencies that exist in the data.

To pool information across the search tokens we need to have them *communicate* with each other. This can be achieved using self-attention which in turn points to a transformer method like DETR [5, 14, 32, 40, 69, 73]. These models utilize fixed *object queries*, which they learn to regress into the predicted boxes, as well as cross-attention, used to look up relevant features from the image independently across individual queries. The combined architecture can utilize global features, which becomes increasingly more useful as the objects' sizes decrease.

Regarding positional dependencies, we show how the diffusion noise affects the matchings between predicted and target boxes. In essence, most approaches like Deformable-DETR [73] start with fixed (x, y) reference points, look up the image features in those locations, and output corrections which are subsequently applied to them. But when diffusion is applied on the initial reference points, they become no longer associated with the object queries, preventing the model from using positional information. To address this challenge we introduce *object query interpolation* as a simple way to learn positional relations for DETR-like models even in the presence of noise over the references.

The resulting generative model can refine its predictions, trade-off accuracy and compute, operate with a different number of search tokens at train and test time, and yields results comparable or better than those of battle-tested deterministic models. Furthermore, it has similarities to particle methods from which ideas like particle pruning and refinement can be borrowed.

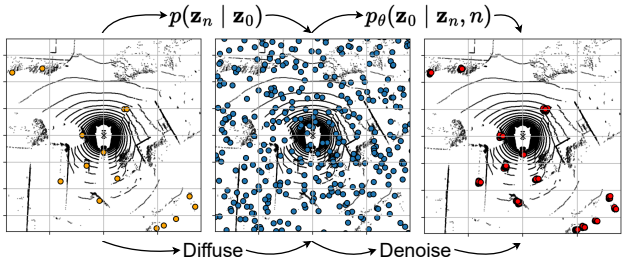

Figure 2. **Diffusion in BEV**. Our approach diffuses the ground-truth object centers in BEV and learns to denoise them. At test time, we start from random references corresponding to the box centers and progressively refine them to their true locations.

Contributions. Our contributions are the following:

1. In Section 3.3 we provide a novel view on the assignment instability problem that is present in most DETR-like models [5, 73] by showcasing how the stochasticity of the diffusion process affects the assignments.
2. In Section 3.4 we showcase our module called *query interpolation* which allows the model to learn positional information in the presence of diffusion noise.
3. We integrate the proposed module into a deformable-DETR [73] variant, called Particle-DETR, which uses diffusion to denoise box centers to their true positions. We further provide a detailed analysis of the performance of the model on the realistic and large-scale NuScenes dataset [4].

2. Related Work

Diffusion-based object detection. Utilizing a diffusion model for detection started with DiffusionDet [8], where the model learns to denoise axis-aligned 2D boxes in the images. First, a backbone network [24, 41], extracts multi-scale image features. At train time, random noise is added to the ground-truth (GT) boxes according to a diffusion schedule, while at test time random boxes are sampled from a Gaussian distribution. Subsequently, a decoder with a region-of-interest-based (ROI) architecture [21, 23] aggregates the features inside each box and produces corrections to the box parameters. The output boxes are then matched to the GT boxes for training.

Other applications. Inspired by DiffusionDet, the usage of diffusion models for other prediction tasks has increased. It has been applied to the denoising of BEV features [75], to prediction of future discrete BEV tokens [70], to action segmentation in videos [38], to weakly supervised object localisation (WSOL) [72], to human motion prediction [1] and pose estimation [18], to domain adaptive semantic segmentation [46], to video anomaly detection [57], to camouflaged object detection [9], to text-video retrieval [30], and to open-world object detection [64].

The DETR family of models. Current object detection in BEV is dominated by DETR-variants [5, 7, 14, 32, 40, 68,

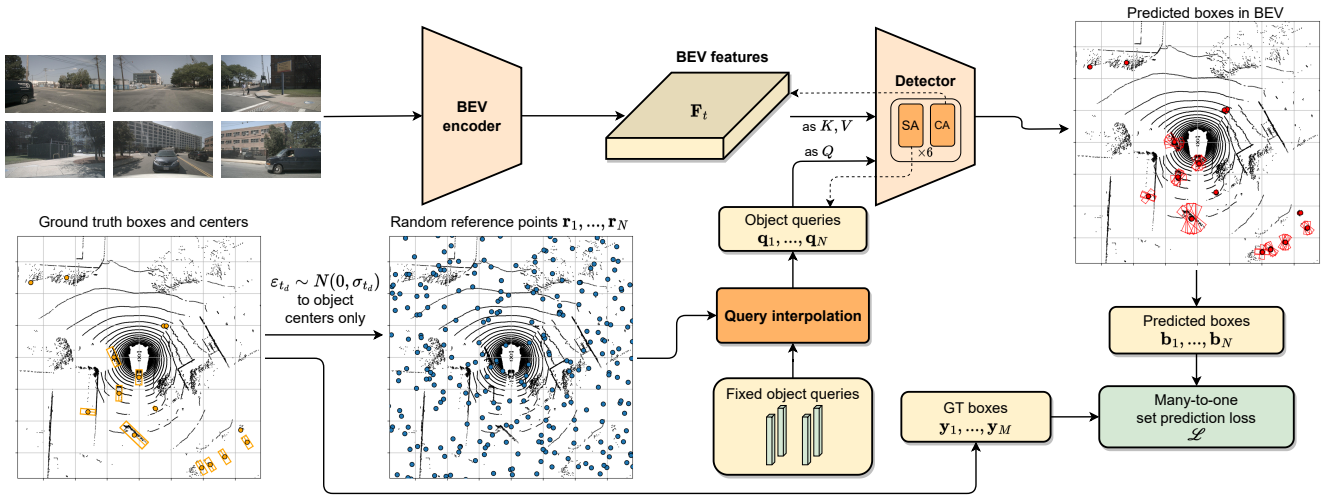

Figure 3. **Diffusion in BEV**. A feature extractor processes the images from the camera feed at the current timestamp. An encoder learns to project these features from the perspective of the car to the top-down orthographic bird-eye-view. At training time we add noise to the ground-truth object centers according to a diffusion schedule, while at test time we directly sample the references. Using our query interpolation module each reference position is associated with a spatial feature. Using these features, the decoder learns to denoise the reference points to their true locations. A set prediction loss is adopted for training.

69]. They utilize a transformer sequence where a fixed number of object queries look up the relevant image features using cross-attention and are transformed into the output boxes. A set-matching step is used to assign predictions to targets. This matching has been described as *unstable*, due to how one prediction can be matched to different targets across the training iterations on the same image. Various approaches mitigate this issue by introducing query denoising [32], where some queries are matched to their target by index, and contrastive denoising [69] where both positive and negative examples are used in each query group.

BEV perception. Transforming the camera features to BEV is an active area of research. It has been done using both traditional approaches where 3D voxels are projected onto the image plane and the image features within the projection are average-pooled [52], or where a categorical depth distribution is estimated for each image pixel, after which the features are "lifted" in 3D according to their depths [48, 50]. Implicit projection, where depth is not estimated explicitly, can be achieved by utilizing self-attention to look up the past BEV and cross-attention to look up the current image features [28, 29, 34, 49, 67, 71]. This is the approach we rely on in this work. Once in BEV, models may perform joint detection and trajectory prediction [15, 26, 27, 42, 65, 71], BEV segmentation [47], tracking [22], or agent interaction analysis [6, 12].

3. Approach

In this section we motivate our method by considering the unique challenge arising when combining diffusion with perception in BEV, cf. Subsection 3.2, and how our method alleviates this challenge, cf. Subsection 3.4.

3.1. Preliminaries

Diffusion models. The goal of diffusion models is to learn to sample from the distribution over a sample space. To that end, as part of the training procedure, a stochastic process adds noise to each input sample according to a predefined schedule. At training time, the model learns to predict the added noise, while at test time one generates initial noise which the model iteratively denoises until a data point from the training distribution is formed.

The *forward* process, which adds noise to the sample at training time, is defined as

$$q(\mathbf{z}_{t_d}|\mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_{t_d}|\sqrt{\bar{\alpha}_{t_d}}\mathbf{z}_0, (1-\bar{\alpha}_{t_d})\mathbf{I}\right), \quad (1)$$

where t_d is the time-index of the diffusion process (different from the temporal frame index t_f in the context of the BEV sequences), $\mathbf{z}_{t_d}$ is the noisy sample at that time, $\mathbf{z}_0$ is the noise-less ground-truth sample, and $\bar{\alpha}_{t_d} = \prod_{s=0}^{t_d} \alpha_s = \prod_{s=0}^{t_d}(1-\beta_s)$ is the corresponding parameter from the schedule controlling the variance of the noise.

The network output $f_\theta(\mathbf{z}_{t_d}, t_d)$ is conditioned on the noisy sample $\mathbf{z}_{t_d}$, the diffusion time t_d and its parameters θ are optimized to minimize the loss

$$\mathcal{L} = \frac{1}{2}\|f_\theta(\mathbf{z}_{t_d}, t_d) - \mathbf{z}_0\|^2. \quad (2)$$

Since this corresponds to a denoising process, at test time we sample random noise $\mathbf{z}_T$ and progressively refine it by

feeding the previous output as the subsequent input to the network, i.e. $\mathbf{z}_0 = f_\theta(f_\theta(...(f_\theta(f_\theta(\mathbf{z}_T, T), T-1))...), 0)$. Various improvements exist to speed-up this process at inference time [10, 44, 45, 56].

Since the noise added to each data sample is independent across all sample elements, we can use this process to generate different objects like images [53], bounding boxes [8], camera poses [59]. Here, the diffusion is applied over box centers (c_x, c_y) in BEV, to which we refer as *particles*.

DETR models. DETR models for object detection [5] rely on a transformer-based architecture. A feature extractor, usually convolutional, extracts image features which are then passed to a transformer encoder where each feature patch can attend to other feature patches. Subsequently, a transformer decoder, relying on a fixed number N of latent vectors $\{\mathbf{q}_1, ..., \mathbf{q}_N\}$ called *object queries*, looks up the features from the encoder and outputs bounding boxes. A one-to-one matching step using the Hungarian algorithm is required to assign predictions to box targets.

The object queries are learned using gradient descent and are fixed at test time. Since positional encodings for the object queries are also used, the model can learn information related to the order of the object queries.

An important modification to the this setup is given by DeformableDETR [73] where $\{\mathbf{q}_1, ..., \mathbf{q}_N\}$ are not only ordered between them, but each object query $\mathbf{q_i}$ is tied to a particular 2D position $\mathbf{r}_i$, called a reference point within the image coordinate frame. Since both the object queries and reference points are learned, the model can focus not only on the content of the pixels, but also on the query positions.

Fixing the reference points $\{\mathbf{r}_1, ..., \mathbf{r}_N\}$ makes training easier because query $\mathbf{q}_i$ will always have the same relative location compared to query $\mathbf{q}_j, j \neq i$. In that case, the cross-attentions in the decoder learn only how to attend to the surrounding features which makes learning more stable. This fixed nature is crucial in relation to the stochasticity we will introduce by the diffusion process.

3.2. Adding diffusion to BEV

Our setup is shown in Figure 3. A feature extractor [24] along with a feature neck [35] processes all camera images from the current timestep t_f, outputting multi-scale feature maps for each camera view. A BEV encoder, in practice BEVFormer [34], projects these features around the ego-vehicle. In BEV, we add noise to the ground-truth object centers and concatenate with additional random locations. These are passed as reference points to the decoder which, similar to DeformableDETR [73], refines some of them into the GT positions.

At test time, we sample initial random box centers and run them through the decoder. Since the model is trained to work with variable reference points, it can plug in the predicted box centers as input reference points in the next denoising step. This allows iterative refinement of our predictions - something that deterministic models like DeformableDETR [73] cannot do because they rely on object queries which are fixed to particular positions.

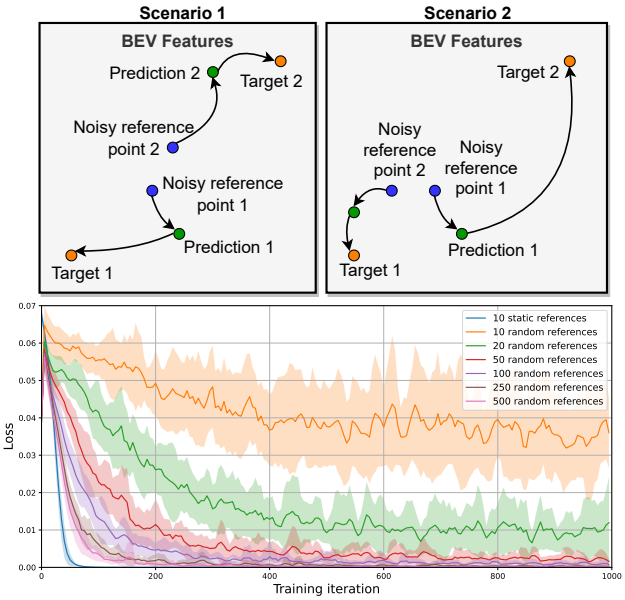

Figure 4. **Label ambiguity**. A case where the random reference points may produce different targets, depending on the matching. The top row shows ambiguity when we match predictions to GTs by total distance (linear sum assignment). The columns show how different samplings of the blue points may push the same feature location in BEV to different targets, thus confusing the model. In all cases the matching is done between the predictions and the targets. The bottom row shows how the training loss depends on the number of references for a simple toy task (cf. suppl. materials).

We follow DETR [5] in applying auxiliary losses to each decoder layer, instead of just the final one. We refer to each decoder layer as a *stage* and to each pass through all decoder layers as a single DDIM [56] step. By having multiple such steps we can trade-off accuracy and compute. Each DDIM step requires evaluating only the decoder.

3.3. Matching

The matching cost used in object detectors from the DETR [5] family typically considers both the predicted box dimensions *and* the predicted class logits. As a result, one cannot say that predictions spatially closer to the GT box will *always* be matched and those farther away will not. Yet, deterministic detectors do converge because even if the matching changes across iterations, the static nature of the inputs – object queries starting from fixed positions – allows one to learn the spatial relationships in the image.

Label ambiguity. In the diffusion case there exist specific situations where learning is, in fact, *impossible* due to the same BEV feature having different targets, depending on

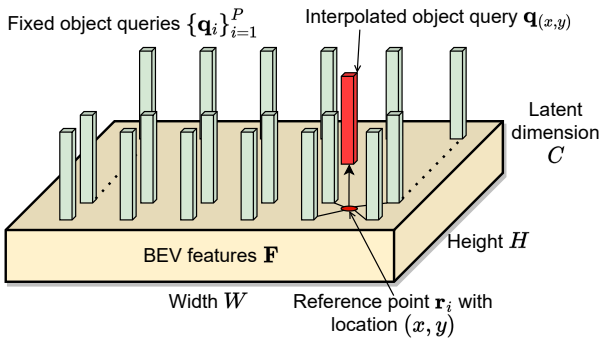

Figure 5. **Query interpolation**. Our method learns a number of regularly-placed object queries, shown in green, which are fixed at test time. This allows the content of each object query to depend on its position in BEV. To accommodate the stochasticity required by the diffusion process we interpolate the object queries at the noisy locations, shown in red. This ensures that if we sample the same reference point many times, we will always obtain the same object query at that location.

the noisy sampling of the reference points. Figure 4 illustrates these conceptually.

Suppose we use the Hungarian algorithm for matching and we sample the initial reference points as the blue points on the top-left plot in Figure 4. Then the matching will be as shown by the arrows. However, if one of the points is sampled differently, as in the top-right plot, we may end up matching them differently. In reality, the BEV features corresponding to the (x, y) position where noisy reference point 1 is, will be pushed by the optimization in the first case toward target 1 and in the second case toward target 2. This creates label ambiguity arising *specifically* due to the random sampling of the starting locations.

Using more object queries than GT boxes reduces the possibility of this ambiguity to hinder the training. This is because having more predictions and matching them with any strategy that takes the distance into account (unlike e.g. matching by index) will make the model produce smaller refinements to the starting reference points. Thus, a high amount of object queries is needed both to detect many objects, but also to help detect them accurately. Explanations on a toy example can be found in the suppl. materials.

3.4. Object query interpolation

Our diffusion is applied on the reference points $\mathbf{r}_i$. As a first approach, we consider a DeformableDETR [73] architecture with N learnable object query vectors which are assigned to their references by index. Thus, object query $\mathbf{q}_i$ may be placed in different (x, y) locations depending on the sampling. While this approach works fairly well in practice, it clearly prohibits the model from learning positional information for query $\mathbf{q}_i$ simply because its position keeps changing during each training iteration.

Instead, we propose to *interpolate* the learned queries at the random sampled reference locations, as shown in Figure 5. We learn a grid of regularly-placed object queries, which we bilinearly interpolate at the reference points. This ensures that sampling the same location $\mathbf{r} = (x, y)$ will always yield the same object query $\mathbf{q}_{(x,y)}$. This also decouples the number of object queries at training and test time, since at training time one needs to learn N queries, but at test time they can be interpolated at N_{test} different locations.

In principle, one can also directly interpolate the BEV at the sampled locations, avoiding the use of learned object queries altogether. Our preliminary experiments showed that learning becomes prohibitively difficult in this case, owing to the diversity and nature of the BEV features. Training is considerably easier if the model looks up the BEV features using cross-attention instead of starting from the BEV features as object queries.

3.5. Additional method components

Loss function. The stochastic nature of the algorithm makes training very slow and difficult if we match predictions and ground truths in a one-to-one manner. To alleviate this, we employ many-to-one matching where many predictions are matched to each GT box. This speeds up training tremendously at the cost of having to post-process the predictions using non-maximum suppression (NMS).

Our loss function is given by

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{reg}, \tag{3}$$

where $\mathcal{L}_{cls}$ is the focal loss between predicted and target class probabilities [36] and $\mathcal{L}_{reg}$ is the ℓ_1 loss between the predicted and GT box parameters. We do not employ a generalized IoU loss [51]. The matching cost is the same as the loss function. For detection, the box parameters include the box center and dimensions, orientation, and velocity in the bird-eye-view plane:

$$\mathbf{b} = (c_x, c_y, c_z, w, h, l, \sin\theta, \cos\theta, v_x, v_y). \tag{4}$$

Particle nature. The many-to-one matching is crucial for our approach because it allows the model to learn gradient fields, or *basins of attraction* around each GT box. This aspect, combined with the random reference points, allows us to look at this architecture as a particle DETR model where multiple particles, the references $\mathbf{r}_1, ..., \mathbf{r}_N$, can move freely and are attracted around the GT boxes. Through the self-attention layers, they can communicate similar to how the best location is globally shared in a particle swarm optimization [31]. The DDIM denoising [56] steps then provide opportunities to refine, renew, or prune the particles, based on their confidence. Additionally, the number of particles which end up on top of a target object can provide a rudimentary measure about the uncertainty of our perceptions at that BEV location. We cannot refer to the

Setting	mAP ↑	NDS ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
1. DiffusionDet [8] - ROI arch., box tokens	0.3846	0.4580	0.7420	0.3382	0.4475	0.5935	0.2222
2. 1 + positional encodings	0.3852	0.4717	0.7357	0.2777	0.5040	0.5008	0.1911
3. DETR arch., random references	0.3929	0.4975	0.7172	0.2707	0.3835	0.4224	0.1963
4. 3 + *simple* n-to-1 matching & NMS	0.4001	0.5203	0.6913	0.2710	0.3415	0.3540	0.1938
5. 3 + simOTA matching & NMS	0.3817	0.5138	0.6444	**0.2641**	**0.3208**	**0.3422**	0.1989
6. 4 + radial suppression	0.4082	0.5203	0.6456	0.2704	0.3528	0.3739	0.1956
7. 4 + training with added fixed queries	0.4077	0.5215	0.6453	0.2696	0.3405	0.3747	0.1935
8. 7 + radial suppression	**0.4088**	**0.5222**	**0.6437**	0.2696	0.3395	0.3768	0.1922
FCOS3D [62]	0.3430	0.4150	0.7250	0.2630	0.4220	1.292	0.153
PGD [61]	0.3690	0.4280	0.6830	0.2600	0.4390	1.2680	0.1850
DETR3D [63]	0.3460	0.4250	0.7730	0.2680	0.3830	0.8420	0.2160
BEVFormer [34], permuted queries	0.3976	0.5073	0.6809	0.2744	0.3722	0.3908	0.1962
BEVFormer, random reference points	0.2997	0.4474	0.735	0.2765	0.3974	0.4179	0.1975
BEVFormer, deterministic	0.4154	0.5168	0.6715	0.2738	0.3691	0.3967	0.1981
BEVFormer-Enh (ours)	**0.4189**	**0.5298**	**0.6319**	**0.2684**	**0.3283**	**0.3737**	**0.1945**

Table 1. **Model progression and results on the NuScenes `val` set**. We showcase how the model components and different architectures affect performance. Models numbered 1-8 are all evaluated with 3 DDIM steps and 900 queries.

search tokens of DETR models [5, 73] as dynamic because they are fixed and do not allow for sequential refinement.

4. Experiments

NuScenes dataset. We evaluate our approach on the large-scale NuScenes dataset [4], comprising almost 1.4 million annotated 3D bounding boxes, across 1000 scenes. There are 10 semantic classes for evaluation. The main metrics of interest are the Mean Average Precision (mAP) and, more importantly, the NuScenes Detection Score (NDS).

For the mAP, detections are calculated by greedily assigning predictions to targets only based on the distance between the predicted and GT centers. There are four distance thresholds - 0.5, 1, 2, and 4 meters. The mAP is calculated as the average precision over 100 recall percentiles and is further averaged over all 10 detectable classes and over these 4 distance thresholds.

Once the predicted boxes are assigned to the targets, one can calculate various true positive metrics – translation error (mATE), scale error (mASE), box orientation (mAOE), velocity (mAVE), attribute error (mAAE) – over the matched pairs. These are weighted together with the mAP to form the NDS metric, which is more realistic in terms of real-life driving performance than the mAP [55].

4.1. Comparison with baselines

We compare against the following relevant models:

1. DiffusionDet [8], which we modify minimally and utilize directly in BEV as our main baseline,
2. DeformableDETR [73], as used in BEVFormer [34], a strong deterministic detector used in BEV.

Baseline. Table 1 shows our main results. We rely on BEVFormer's encoder to project the images into the top-down view. Since the original DiffusionDet works only on axis-aligned boxes, we modify it by adopting rotated ROI pooling similar to [66]. The architecture follows a six stage RCNN [21] decoder where each stage takes the BEV features and a number of rotated boxes in BEV, parameterized as (c_x, c_y, w, h, θ). The BEV features falling into the rotated box are aggregated and deformable convolutions [13] are applied to model instance interactions between different boxes. Each stage outputs corrections which are applied to the current boxes to produce the subsequent-stage boxes. Overall, applying DiffusionDet directly in BEV yields good performance compared to reference models [61–63] but inferior compared to the deterministic BEVFormer.

Positional encodings. It is common to encounter certain classes more often in some positions relative to the ego-vehicle, e.g. pedestrians appear in front of the car less often than at the side of the car. To force the ROI-based architecture to consider the absolute locations of the boxes in BEV, we use sinusoidal positional encodings [58], which we concatenate to the aggregated BEV features for each box token.

Global features to address sparsity. ROI-based architectures emphasize the local features inside each box. Such a prior may be sufficient on some datasets [37], but for smaller objects more global features are needed. This motivates us to consider a DETR-based architecture where instead of boxes and ROI-pooling we have object queries and attention. Now, each stage first applies self-attention over the object queries, thereby considering their relative position and content, and then applies cross-attention over the BEV, which has potentially unlimited view and can aggregate more global BEV features for each token.

Many-to-one matching. With random reference points $\mathbf{r}_1, ..., \mathbf{r}_N$, the supervisory signal when matching in a one-

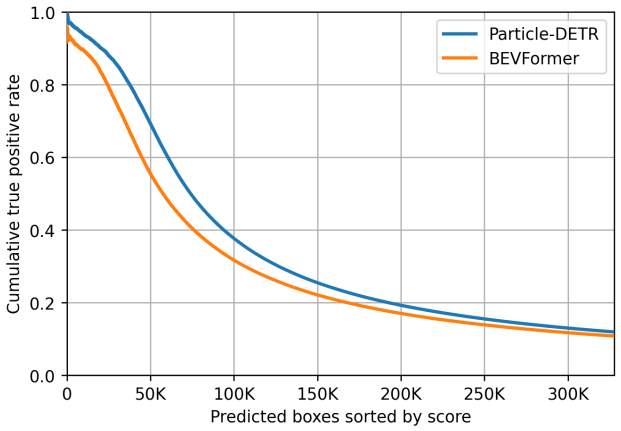

Figure 6. **Sparsification plot**. Compared to BEVFormer, our Particle-DETR makes more confident predictions early on which yields higher precision when detecting.

to-one fashion is fairly weak. Thus, we experimented with two many-to-one matching strategies. The first we call *simple N-to-1* because it simply repeats the GT boxes a number of times, stacking them on top of each other, and then applies the linear sum assignment solver for matching. For the second strategy we use the SimOTA [20] approximation to optimal transport assignment [19], which matches a variable number of predictions to each target.

Detection accuracy. The results show that our diffusion-based Particle-DETR achieves good performance and noticeably outperforms the baseline DiffusionDet [9] on both mAP and NDS. Even more, its performance is comparable to that of deterministic approaches like BEVFormer [34]. Our generative approach achieves higher NDS, showing that once a detection is established, the predicted box dimensions, orientation, and velocity are more accurate. Additionally, our Particle-DETR is better calibrated compared to BEVFormer since more confident predictions have a higher cumulative detection rate, as shown in Figure 6.

Enhancement with static references. The random references allow the model to learn basins of attraction around each GT center. However, nothing prevents us from utilizing fixed references as well, which yield higher precision. Thus, we further experiment with a setup in which we train with two sets of references - one random, coming from the diffusion process, and one static. In turn, the two reference sets result in two sets of queries - one where the queries are interpolated at the random locations (cf. Subsection 3.4), and one where the queries are learned and fixed, as in [34, 73]. The joint training captures any synergies between the random and fixed queries, improving the performance of both. At test time, to keep the number of queries comparable to previous models, we can use only one query set. Using the diffusion queries we obtain our final Particle-DETR model. Using the static ones we obtain an enhanced BEVFormer which we call BEVFormer-Enh.

4.2. Implementation details

The implementation of our Particle-DETR is straightforward following BEVFormer [34]. We train the model for the same number of iterations as BEVFormer and the number of parameters is similar. The training details with pseudocodes for the train-test behaviour can be found in the supplementary materials.

Gradient detachment. To further facilitate training, we equip each decoder layer with *look forward twice* updates [69], where the reference points for each decoder layer are not detached from the computation graph when computing the next-layer reference points during the forward pass.

Filtering of predictions. At training time, the many-to-one matching helps to learn the basins of attraction around each GT center. However, at test time this induces behaviour where multiple predictions stack on top of each other. Thus, to avoid additional false positives, we employ NMS and also utilize a small score threshold which filters any predictions with confidence below it.

Radial suppression. We found that very small objects like traffic cones do not overlap and are missed by NMS. For that reason, we introduced *radial suppression* to further filter out the boxes. In essence, we first order the predictions by decreasing confidence. Then we replace the most confident ones with weighted averages of their close-by boxes which, in turn, are filtered:

$$\mathbf{b}_i = \frac{\sum_k \mathbf{b}_k \pi_k}{\sum_k \pi_k}, \forall k : \sqrt{(c_{x,i} - c_{x,k})^2 + (c_{y,i} - c_{y,k})^2} < r. \tag{5}$$

Here $c_{x,k}$ is the x-coordinate of the center of the k-th box, and π_k is the confidence for that box. We implement radial suppression independently for each semantic class.

4.3. Additional properties

Flexibility. The architecture of our Particle-DETR allows us to train it with one number of queries but evaluate with a different number. Additionally, the number of DDIM

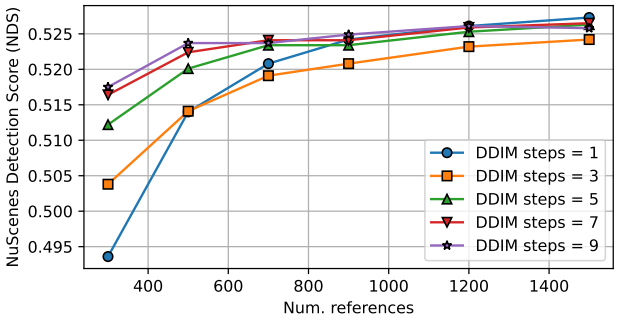

Figure 7. **Effect of number of references on NDS**. Holding the number of DDIM steps fixed, NDS increases as the number of random references increases.

steps [56] allows us to further trade-off accuracy and compute. Figure 7 shows that both increasing the number of DDIM steps and the number of particles used improves performance. With 900 references it only takes a single DDIM step to outperform BEVFormer on NDS.

Stochastic nature of results. Since we rely on randomly sampled initial reference points, the outputs of our method are stochastic. Table 2 shows statistics over 10 test runs. Performance is very consistent across them.

4.4. Qualitative study

Here we perform a qualitative comparison between our predictions and those of BEVFormer [34]. In general, the higher NDS which results from the diffusion process makes our detections more precise in terms of location, size, and orientation, which can be particularly beneficial for very small objects (e.g. traffic cones) near the car. On some scenes our method recognizes even partially-occluded objects earlier and more confidently, as shown in Figure 8.

It is common for models to struggle with accurate estimation of the dimensions of large objects like buses and trucks. This is because they obscure the camera's field of view considerably, making it hard to estimate where the object ends. We notice that in some scenes our method improves noticeably in this regard. Further visualizations and analysis can be found in the supplementary materials.

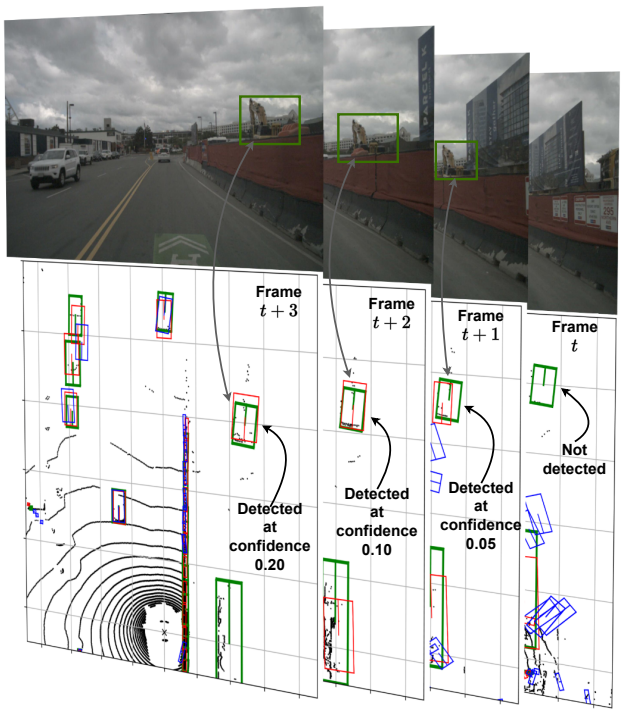

Figure 8. **Sample predictions in BEV**. Green boxes are ground-truths, red are predicted by our Particle-DETR, and blue is predicted by BEVFormer, compared to which we detect earlier in the frames and more confidently, even for less-common objects.

Metric	BEVFormer	Ours (stoc.)	Ours
NDS ↑	0.5168	0.5271 (0.0002)	**0.5287**
mAP ↑	0.4154	0.4163 (0.0003)	**0.4184**
mATE ↓	0.6715	0.6415 (0.0008)	**0.6386**
mASE ↓	0.2738	0.2689 (0.0002)	**0.2686**
mAOE ↓	0.3691	0.3390 (0.0009)	**0.3362**
mAVE ↓	0.4179	**0.3688** (0.0010)	**0.3688**
mAAE ↓	0.1981	**0.1920** (0.0006)	0.1931

Table 2. **Performance statistics on the NuScenes `val` set**. We compare our stochastic Particle-DETR (col. 3), evaluated with 1500 queries and 1 DDIM step, and the deterministic BEVFormer-Enh (col. 4) to the original BEVFormer. The standard deviations for the random methods are shown in parentheses.

5. Discussion

Precision in generative models. Compared to text-to-image tasks which tolerate a large amount of variation in the generated samples, object detection requires precision in the outputs. Hence, adjusting for the number of queries, we find a small performance gap in mAP with respect to deterministic approaches natural, as the random reference inputs will always induce a distribution on the outputs.

Uncertainty. One benefit of learning a distribution over the boxes is that this provides a rudimentary way to understand their uncertainty. Unfortunately, it is likely that it mixes epistemic uncertainty resulting from the estimated model parameters and aleatoric uncertainty related to the randomness of the boxes themselves. We show heatmaps for the box distributions in the suppl. materials.

Temporal modeling. We acknowledge that methods like StreamPETR [60], SparseBEV [39], or HoP [74] outperform BEVFormer by means of more sophisticated designs. They emphasize the importance of extended temporal modeling of the objects, while the focus in this work is different - to investigate how noisy 2D reference points can be used together with dense BEV features. We expect our method to be complementary to other such alternatives which use dense BEV features, like HoP [74].

6. Conclusion

We have shown that naively using previous generative approaches for BEV detection yields a performance gap. To close it, we adopt a transformer-based architecture and a specific query interpolation module to facilitate the model in learning positional information even in the presence of diffusion. Applying diffusion over particles yields a unique interpretation to our approach based on particle methods. We greatly improve on previous generative methods and achieves comparable results to strong deterministic ones.

References

[1] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2317–2327, 2023. 2

[2] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 1

[3] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 135–152. Springer, 2020. 1

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 6, 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 4, 6

[6] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 3

[7] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6633–6642, 2023. 2

[8] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023. 2, 4, 6

[9] Zhennan Chen, Rongrong Gao, Tian-Zhu Xiang, and Fan Lin. Diffusion model for camouflaged object detection. *arXiv preprint arXiv:2308.00303*, 2023. 1, 2, 7, 4, 5

[10] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12413–12422, 2022. 4

[11] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[12] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7801–7807. IEEE, 2023. 3

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 6

[14] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2021. 2

[15] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? *Advances in Neural Information Processing Systems*, 35:15657–15671, 2022. 3

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 2

[18] Runyang Feng, Yixing Gao, Tze Ho Elden Tse, Xueqing Ma, and Hyung Jin Chang. Diffpose: Spatiotemporal diffusion model for video-based human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14861–14872, 2023. 2

[19] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021. 7

[20] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 7, 2, 4

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2, 6

[22] JunYoung Gwak, Silvio Savarese, and Jeannette Bohg. Minkowski tracker: A sparse spatio-temporal r-cnn for joint object detection and tracking. *arXiv preprint arXiv:2208.10056*, 2022. 3

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 4

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[26] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision, pages 15273–15282, 2021. 3

[27] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 3

[28] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 3

[29] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1042–1050, 2023. 3

[30] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. Diffusionret: Generative text-video retrieval with diffusion model. *arXiv preprint arXiv:2303.09867*, 2023. 2

[31] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995. 5

[32] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2, 3

[33] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022. 1

[34] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 1, 3, 4, 6, 7, 8, 2, 5

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 6

[38] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10139–10149, 2023. 2

[39] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18580–18590, 2023. 8

[40] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2

[41] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[42] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018. 3

[43] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, pages 1–55, 2023. 1

[44] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5517–5526, 2023. 4

[45] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 4

[46] Duo Peng, Ping Hu, Qiuhong Ke, and Jun Liu. Diffusion-based image translation with label guidance for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 808–820, 2023. 2

[47] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5935–5943, 2023. 3

[48] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 1, 3

[49] Zequn Qin, Jingyu Chen, Chao Chen, Xiaozhi Chen, and Xi Li. Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8690–8699, 2023. 3

[50] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021. 3

[51] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 5, 2

[52] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 1, 3

[53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 4

[54] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[55] Tim Schreier, Katrin Renz, Andreas Geiger, and Kashyap Chitta. On offline evaluation of 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4084–4089, 2023. 6

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 4, 5, 8, 2

[57] Anil Osman Tur, Nicola Dall'Asen, Cigdem Beyan, and Elisa Ricci. Exploring diffusion models for unsupervised video anomaly detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2540–2544. IEEE, 2023. 2

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

[59] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 4

[60] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023. 8

[61] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, pages 1475–1485. PMLR, 2022. 6

[62] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 6

[63] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 6

[64] Yanghao Wang, Zhongqi Yue, Xian-Sheng Hua, and Hanwang Zhang. Random boxes are open-world object detectors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6233–6243, 2023. 2

[65] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020. 3

[66] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3520–3529, 2021. 6, 1

[67] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023. 3

[68] Mingqiao Ye, Lei Ke, Siyuan Li, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Cascade-detr: Delving into high-quality universal object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6704–6714, 2023. 2

[69] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 2, 3, 7

[70] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*, 2023. 2

[71] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 3

[72] Yuzhong Zhao, Qixiang Ye, Weijia Wu, Chunhua Shen, and Fang Wan. Generative prompt model for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6351–6361, 2023. 2

[73] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 4, 5, 6, 7

[74] Zhuofan Zong, Dongzhi Jiang, Guanglu Song, Zeyue Xue, Jingyong Su, Hongsheng Li, and Yu Liu. Temporal enhanced training of multi-view 3d object detector via historical object prediction. *arXiv preprint arXiv:2304.00967*, 2023. 8

[75] Jiayu Zou, Zheng Zhu, Yun Ye, and Xingang Wang. Diffbev: Conditional diffusion model for bird's eye view perception. *arXiv preprint arXiv:2303.08333*, 2023. 2