MonoDiff: Monocular 3D Object Detection and Pose Estimation with Diffusion Models

Yasiru Ranasinghe, Deepti Hegde, and Vishal M. Patel
Johns Hopkins University, Baltimore, USA
{dranasi1, dhegde1, vpatel36}@jhu.edu

Abstract

3D object detection and pose estimation from a single-view image is challenging due to the high uncertainty caused by the absence of 3D perception. As a solution, recent monocular 3D detection methods leverage additional modalities, such as stereo image pairs and LiDAR point clouds, to enhance image features at the expense of additional annotation costs. We propose using diffusion models to learn effective representations for monocular 3D detection without additional modalities or training data. We present MonoDiff, a novel framework that employs the reverse diffusion process to estimate 3D bounding box and orientation. But, considering the variability in bounding box sizes along different dimensions, it is ineffective to sample noise from a standard Gaussian distribution. Hence, we adopt a Gaussian mixture model to sample noise during the forward diffusion process and initialize the reverse diffusion process. Furthermore, since the diffusion model generates the 3D parameters for a given object image, we leverage 2D detection information to provide additional supervision by maintaining the correspondence between 3D/2D projection. Finally, depending on the signal-to-noise ratio, we incorporate a dynamic weighting scheme to account for the level of uncertainty in the supervision by projection at different timesteps. MonoDiff outperforms current state-of-the-art monocular 3D detection methods on the KITTI and Waymo benchmarks without additional depth priors. MonoDiff project is available at: https://dylran.github.io/monodiff.github.io.

1. Introduction

Research on monocular 3D object detection is currently a focal point in various fields, such as autonomous driving [8, 36], robotic navigation [29, 44], and beyond [32]. The objective is to generate 3D bounding box parameters based on the identification of objects in 2D images [8, 43, 64, 72].

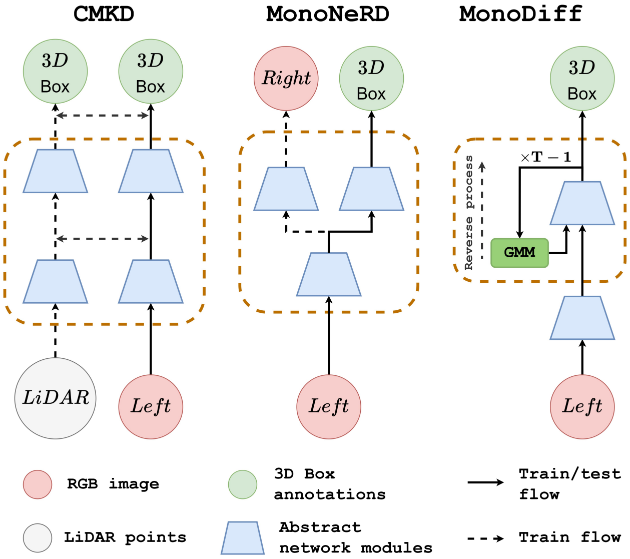

Figure 1. Comparison between existing architectures for effective representation learners for monocular 3D object detection. CMKD [28] uses LiDAR data, and MonoNeRD [74] estimates the stereo and depth image during training. MonoDiff uses monocular images with diffusion models as effective representation learners.

Previous studies have extracted 3D information for object poses using 2D/3D constraints and geometric priors. These constraints typically necessitate additional annotations [8, 24] or the employment of Computer-Aided Design models [5, 47]. Alternatively, some earlier approaches use pseudo-LiDAR from depth estimates [50, 65, 71] or integrate image features with depth maps as a precursor for the 3D detection model. Lately, the monocular 3D detection research has focused on generating corresponding bird's-eye-view (BEV) representations [11, 53, 77, 78] from 2D images to work with pre-trained 3D detectors.

Following the improvement from generating meaningful representations, recent methods have demonstrated the effectiveness of leveraging additional training data or modalities, as illustrated in Fig. 1, for inferring 3D information [25, 38, 64]. While these additional modalities help learn effective representations, their inclusion burdens the cost of data acquisition and annotation. Notably, recently proposed denoising diffusion probabilistic models [26, 66], known as diffusion models, have emerged as proficient representation learners for discriminative tasks [18, 60]. The forward diffusion process in these models is conceptualized as an augmentation technique, contributing to more effective representation learning than a conventional single forward pass network.

Up until recently, diffusion models [26, 66] have exhibited superior performance in learning data distributions for generative tasks, outperforming GANs and achieving state-of-the-art results [2, 15]. Capitalizing on their success in generative tasks, diffusion models have found application in various image-to-image applications, including super-resolution [62], inpainting [45], image segmentation [1, 2, 22], and image deblurring [59]. Building on the demonstrated effectiveness of diffusion models as representation learners for various computer vision challenges [2], several contemporary approaches have adopted diffusion models for other perception tasks like 2D object detection [6], crowd analysis [57], and human pose estimation [18]. Consequently, we investigate the role of diffusion models in monocular 3D object detection and pose estimation, focusing on their ability to elevate 2D detections to 3D parameters.

MonoDiff conceptualizes the 3D detection and pose estimation of an object through a reverse diffusion process, wherein a distribution characterized by high variance undergoes a progressive transformation towards one marked by low variance. Due to the high variation in bounding box dimension, the uncertainty along different axes will differ. Thus, the estimates will not necessarily converge to the normal distribution (i.e., zero mean and unit variance Gaussian) after completing the forward process. The standard diffusion process is thus not the most appropriate to model the uncertainty or initialize the starting bounding box distribution for 3D localization and pose estimation tasks.

To address this, we model the latent distributions of the reverse process using Gaussian Mixture Models (GMM) to account for different uncertainty levels along different dimensions. Furthermore, we use 2D bounding box information to supervise the localization and orientation estimates from the reverse diffusion process using the corresponding constraints of 3D-2D projection. Lastly, we use a signal-to-noise ratio-based weighting scheme on 2D/3D projection supervision to account for the uncertainty levels at different timesteps.

The contributions of the paper are:

- We present MonoDiff, a novel detection framework that leverages the distribution learning ability of diffusion models to enable accurate 3D perception from a single image.
- We present a GMM-based initialization for the reverse diffusion process instead of a normal distribution to resolve the uncertainty variation along different 3D localization and orientation parameters.
- We experiment on both the KITTI-3D detection benchmark [20] and the Waymo Open Dataset [68]. Our experimental results showcase the effectiveness of MonoDiff, surpassing the state-of-the-art methodologies without additional modalities.

2. Related work

Monocular 3D object detection is designed to establish a transformation between the camera sensor input and 3D attributes, as outlined by Fang et al. [17]. In contrast to stereo methodologies [10, 39, 40, 75] that rely on dual RGB cameras, monocular systems operate with a sole *single-view* input. Initially, Mousavian et al. [48] proposed a technique involving the regression of relatively stable 3D parameters based on 2D detections. More recently, MonoJSG [42] introduces a method that utilizes a joint semantic and geometric cost volume to mitigate the inherent challenges of ill-posedness in monocular 3D object detection. Furthermore, MonoGround [55] suggests incorporating a ground plane as an additional depth estimation source without necessitating extra data or modalities. Recent advancements in monocular 3D detection also integrate geometric properties [13, 43, 80] to effectively address the challenges associated with the ill-posed nature of the task.

Pose estimation within 3D object detection systems is concerned with accurately determining the orientation of instances. Various solutions, both closed-form and iterative, assume correspondences between 2D keypoints in the image and a 3D object model, as discussed in [4, 35]. Alternatively, some approaches involve constructing 3D models for object instances and then identifying the 3D pose in the image that best aligns with the model [19, 61]. Addressing images with multiple instances, architectures akin to Fast-RCNN were utilized in [3, 7, 9, 30, 31], where the region-of-interest features captured instance appearance, and a classification head provided pose predictions.

Pseudo-auxiliary feature-based methods utilize additional data sources or modalities such as LiDAR, BEV, and stereo information during training to establish priors for the detection model. Pseudo-LiDAR-based 3D detectors [11, 46, 54, 71] derive benefits from both emulating the representation of LiDAR data during inference and leveraging the accurate 3D information provided by LiDAR during training. Typically, these methods involve transforming 2D images into intermediate 3D representations, such as pseudo-point clouds through depth estimators [70, 71]. Subsequently, LiDAR-based methods are applied to these representations. Simultaneously, MonoNeRD [74] incorporates stereo images during training and builds a pipeline to

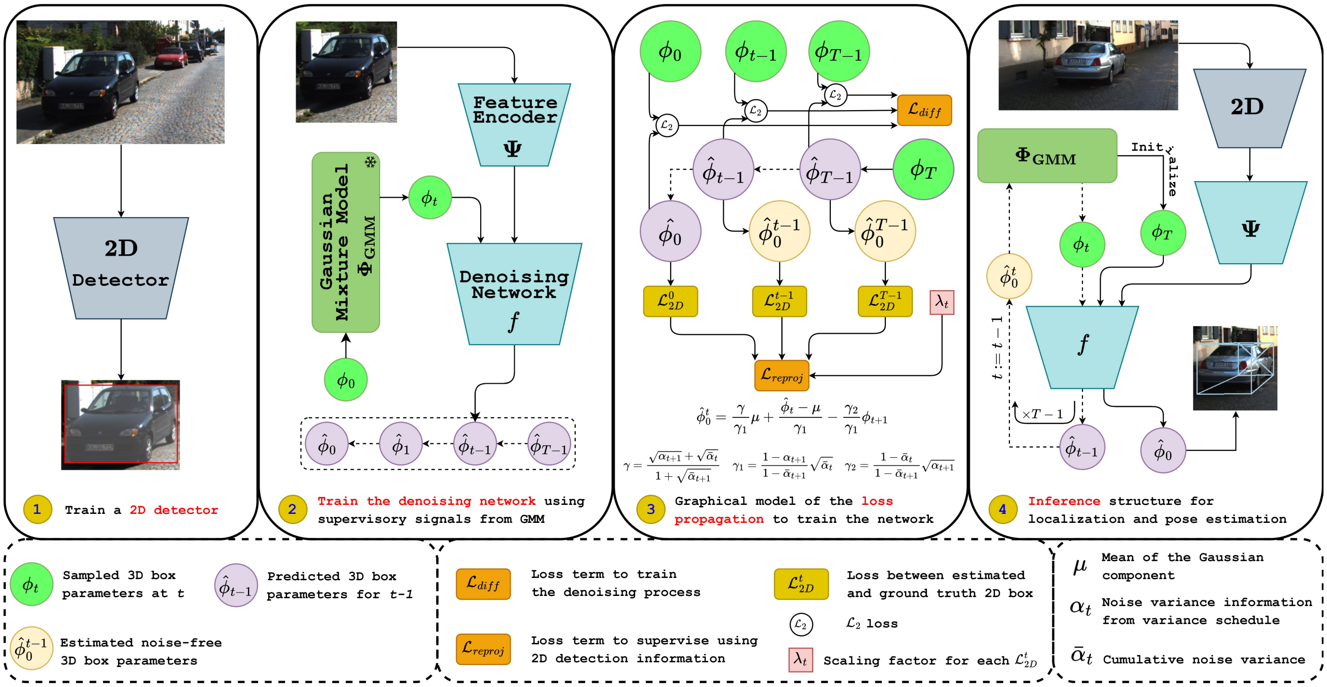
2D
Detector
1 Train a 2D detector
Feature
Encoder
Ψ
Gaussian
Mixture Model
Φ_{GMM}
*
ϕ_t
ϕ_0
Denoising
Network
f
$\hat{\phi}_0$
$\hat{\phi}_1$
$\hat{\phi}_{t-1}$
$\hat{\phi}_{T-1}$
2 Train the denoising network using supervisory signals from GMM
ϕ_0
ϕ_{t-1}
ϕ_{T-1}
$\mathcal{L}_2$
$\mathcal{L}_{diff}$
$\hat{\phi}_{t-1}$
$\hat{\phi}_{T-1}$
ϕ_T
$\hat{\phi}_0$
$\hat{\phi}_0^{t-1}$
$\hat{\phi}_0^{T-1}$
$\mathcal{L}_{2D}^0$
$\mathcal{L}_{2D}^{t-1}$
$\mathcal{L}_{2D}^{T-1}$
λ_t
$\mathcal{L}_{reproj}$
$\hat{\phi}_0^t = \frac{\gamma}{\gamma_1}\mu + \frac{\hat{\phi}_t - \mu}{\gamma_1} - \frac{\gamma_2}{\gamma_1}\phi_{t+1}$
$\gamma = \frac{\sqrt{\alpha_{t+1}} + \sqrt{\bar{\alpha}_t}}{1 + \sqrt{\bar{\alpha}_{t+1}}}$ $\gamma_1 = \frac{1 - \alpha_{t+1}}{1 - \bar{\alpha}_{t+1}}\sqrt{\bar{\alpha}_t}$ $\gamma_2 = \frac{1 - \bar{\alpha}_t}{1 - \bar{\alpha}_{t+1}}\sqrt{\alpha_{t+1}}$
3 Graphical model of the loss propagation to train the network
2D
Φ_{GMM}
Initialize
Ψ
ϕ_t
ϕ_T
$\hat{\phi}_0^t$
$t := t-1$
f
$\times T-1$
$\hat{\phi}_{t-1}$
$\hat{\phi}_0$
4 Inference structure for localization and pose estimation
ϕ_t Sampled 3D box parameters at t
$\hat{\phi}_{t-1}$ Predicted 3D box parameters for t-1
$\hat{\phi}_0^{t-1}$ Estimated noise-free 3D box parameters
$\mathcal{L}_{diff}$ Loss term to train the denoising process
$\mathcal{L}_{reproj}$ Loss term to supervise using 2D detection information
$\mathcal{L}_{2D}^t$ Loss between estimated and ground truth 2D box
$\mathcal{L}_2$ $\mathcal{L}_2$ loss
λ_t Scaling factor for each $\mathcal{L}_{2D}^t$
μ Mean of the Gaussian component
α_t Noise variance information from variance schedule
$\bar{\alpha}_t$ Cumulative noise variance

Figure 2. (1) We train a 2D detector to localize objects on the images. (2) We estimate the parameters of the Gaussian Mixture Model (Φ_{GMM}) to generate supervisory signals ($\phi_1, \phi_2, \cdots, \phi_T$) from the ground truths for the denoising network (f). The localized object features (ψ) are generated using a feature encoder (Ψ) and provided to f as the conditioning. The estimates ($\hat{\phi}_t$) of f are saved at each time step to learn the reverse diffusion process. (3) Loss propagation to train the pipeline. $\mathcal{L}_{diff}$ is computed between $\hat{\phi}_t$ and ϕ_t for $\{T-1, \cdots, 0\}$. At each time step, an estimate $\hat{\phi}_0^t$ for ϕ_0 is produced using $\hat{\phi}_t$ according to the equation to compute the loss ($\mathcal{L}_{2D}^t$) between 2D bounding box information and the projection of estimated 3D bounding box. Then $\mathcal{L}_{2D}^t$ is scaled using λ_t at each time step and accumulated to compute $\mathcal{L}_{reproj}$. (4) During the inference process, objects are first localized and then passed through Ψ and f to estimate the 3D bounding box.

recover the depth map and the right RGB image in addition to the 3D bounding boxes. For that, Xu et al. [74] introduce scene modeling to generate 3D representations akin to Neural Radiance Fields. Across all these approaches, the overarching goal is to produce complementary features using monocular images to enhance 3D object detection and pose estimation. In this study, we leverage diffusion models to extract features at different timesteps, reducing the need for additional training data while augmenting the size of features learned through noise augmentation.

2.1. Diffusion models for perception

Diffusion models have proven to be a potent methodology for learning a data distribution suitable for sampling. Originally introduced DDPMs [66] to generate images or for image-to-image translation, have undergone recent advancements, particularly in terms of improved inference speed [26, 49, 67]. Previous research has delved into applying diffusion models across diverse generative tasks, ranging from image inpainting [45] to text generation [41].

Pioneering the integration of diffusion models into object detection, DiffusionDet [6] addresses the 2D object detection problem by denoising random boxes into object bounding boxes through the diffusion process. Building on this concept, DiffRef3D [33] extends the application of diffusion models to 3D perception tasks, employing LiDAR point clouds instead of images. Additionally, diffusion models have found diverse real-time applications, including human pose estimation [21, 27, 76], crowd analysis [14, 57], and segmentation [56, 73]. In this context, we investigate the utilization of diffusion models to address the challenges of 3D detection and pose estimation within the framework of our MonoDiff.

3. MonoDiff

We aim to generate accurate 3D coordinates and poses with diffusion models. We condition the denoising process on the image features of objects localized by a fixed 2D detector, and the diffusion process is formulated as the iterative noising of a vector with the 3D coordinates and angles into a Gaussian distribution. The components of our proposed MonoDiff pipeline along with loss propagation and inference are illustrated in Fig. 2.

3.1. Background

The diffusion model has two processes: the forward and the reverse process.

The forward process is the approximate posterior $q(x_{1:T}|x_0)$ modeled by a Markov chain that gradually transforms the original data distribution to a normal distribution $\mathcal{N}(0, I)$ by adding Gaussian noise to the original data. At each degradation step, the noise is sampled from a predefined parametrized noise schedule depending on the timestep t. At each step t, the noise is incrementally added to the signal according to

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I).$$

This formulation allows for the sampling of degraded samples at any given timestep in closed form by

$$q(x_t|x_0) := \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right),$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$.

The reverse process in the standard diffusion models, first, a realization is sampled from a normal Gaussian distribution. It is then iterated through the denoising network to transverse to the data distribution. We refer the readers to [26] for more details.

3.2. MonoDiff forward diffusion process

The uncertainty along different dimensions differs in similar object categories, and the geometric structure and orientation of bounding boxes are different across classes. Hence, the target distribution of the forward diffusion process does not have to converge to a normal distribution, i.e., a Gaussian distribution with a zero mean and a unit variance, as the means and variances of different dimensions will differ. If we initialize from a $\mathcal{N}(\mathbf{0}, \mathbf{I})$, this overlooks the difference in variance between the dimensions. Hence, we don't enforce the distribution to converge to $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in the forward diffusion process. Further, using a normal distribution to initialize the inference process is not the most suitable, as a single distribution does not account for these variations. Also, $\mathcal{N}(\mathbf{0}, \mathbf{I})$ initialization represents random boxes without any regard for the object. This is similar to DiffusionDet [6] and DiffBev [79], which trains a separate decoder to predict the box parameters from the image features instead of the denoising network.

In MonoDiff, we use a mixture of Gaussians (GMM) [37] to define the initial bounding boxes' parameters' (dimensions and orientations) distribution, ensuring the compatibility with the Gaussian assumption inherent in the formulation of DDPMs. Utilizing a GMM for initialization offers the advantage of enhancing the inference speed of the network. This is achieved by sampling the starting latent variable from a distribution containing information about bounding boxes instead of a random initialization. Moreover, initializing and noise sampling from the GMM contribute to a more efficient implementation of DDPMs by constraining the range of latent variable values at each timestep. Then, the set of parameters (Φ_{GMM}) is optimized as follows using the Expectation-Maximization algorithm:

$$\underset{\Phi_{GMM}}{\operatorname{argmax}} \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\phi_0^n|\mu_k, \Sigma_k),$$

where ϕ_0^ns are ground truth bounding box parameters, and π_k, μ_k, and Σ_k are the prior, mean, and covariance of the individual Gaussian component.

Subsequently, we degrade the ground truth bounding boxes by adding noise from the GMM-based initialization provided by Φ_{GMM}. To approximate Φ_{GMM} at the end of the forward diffusion process, we modify the forward original DDPM diffusion equation following [23] as follows:

$$\phi_t = \mu + \sqrt{\alpha_t}(\phi_0 - \mu) + \sqrt{(1-\alpha_t)} \cdot \epsilon. \qquad (1)$$

where ϕ_t is a sample from the latent distribution Φ_t at the t^{th} timestep. Then, μ is the mean of the selected Gaussian component, and noise (ϵ) is sampled from a zero mean Gaussian distribution but with covariance equivalent to that of the selected Gaussian component.

Since we approximate the Φ_{GMM} at the end of the forward diffusion process, we denote it as Φ_T. Then, we choose a Gaussian component according to the π_k distribution. Note that, according to Eq. (1), at the end of the forward diffusion process, the sampled point ϕ_T will be from the selected Gaussian component as the effect of ϕ_0 fades away since α_T goes to zero. During training, we can assign the ground truth boxes to the components in the GMM. These assignments are used as additional labels during the training of the 2D detector to give the best cluster assignment during testing.

3.3. MonoDiff reverse diffusion process

In the reverse process, the denoising network is employed to estimate latent samples at each timestep, enabling the determination of 3D coordinates and orientation as a deterministic distribution derived from the initial distribution Φ_T. The optimization of the denoising network involves leveraging intermediate distributions to learn the reverse diffusion process effectively. For a sampled ϕ_t from Φ_t, the denoising network f, conditioned on image features and the diffusion step, reconstructs $\hat{\phi}_{t-1}$ from ϕ_t using the formulation:

$$\hat{\phi}_{t-1} = f(\phi_t, \psi, t), \;\; t \in \{1, ..., T\}.$$

to learn the propagation of distributions.

3.4. 2D Reprojection Regulation

To enhance supervision in 3D localization and pose estimation, we incorporate information from the 2D bounding box of the object of interest. Since the estimated 3D bounding box of the localized object can be projected onto the image plane, the projection should fall within its 2D bounding box. Though these two do not have to overlap, we can expect a snug fit between the projected bounding box and the ground truth bounding box.

To project the 3D bounding box, we need the rotation matrix R and the translation vector T. However, since the diffusion pipeline operates on localized objects, we can assume the T as the origin and shift the 2D ground truth information accordingly. Consequently, the coordinates of the 3D bounding box can be expressed using the dimensions of the bounding box. Next, we estimate the 2D bounding box using the projected 3D bounding box vertices and compare it with the ground truth 2D box. The reprojection provides additional supervision since the projected bounding box is a function of R and D, estimated by the diffusion pipeline.

3.5. Choice of generative parameters

We generate the rotation matrix R and produce the dimension matrix D using the reverse process for parameter estimation of the bounding boxes. To solve for T, we identify the vector that gives the closest projection to the initial 2D bounding box from the detector. More details are provided in the supplementary on solving for translation. The range of values probable for box dimensions is small compared to the translation vector, as the 3D box size does not vary depending on the position of the camera coordinate system. This is desirable for learning the distribution with diffusion models [26].

3.6. Loss function

To build the connections between 3D representations and 2D observations, we optimize the parameters of the denoising network f discussed in Sec. 3.3. The overall objective is expressed as a composite loss function consisting of diffusion loss ($\mathcal{L}_{diff}$) and reprojection loss ($\mathcal{L}_{reproj}$).

Diffusion loss. The denoising network parameters are optimized to produce $\hat{\phi}^i_{t-1}$ from ϕ^i_t in a single forward pass. Unlike the usual objective where the network is trained to estimate the amount of noise in the $\hat{\phi}^i_{t-1}$, we formulate the diffusion loss $\mathcal{L}_{\mathit{diff}}$ as follows:

$$\mathcal{L}_{\mathit{diff}} = \sum_{t=1}^{T}\sum_{i=1}^{N} \|f(\phi^i_t, \psi^i, t) - \phi^i_{t-1}\|^2_2,$$

following preliminary work on DDPMs [26, 67].

Reprojection loss. The reprojection of the 3D bounding box should tightly fit into the 2D detection window, and we ensure this by using the bounding box and IoU losses from the 2D detection network. We compute the loss across all timesteps in the diffusion loss, and the reprojection loss is written as:

$$\mathcal{L}_{\mathit{reproj}} = \sum_{t=1}^{T}\sum_{i=1}^{N} \lambda_{bbox}\mathcal{L}^{ti}_{bbox} + \lambda_{iou}\mathcal{L}^{ti}_{iou},$$

where $\mathcal{L}^{ti}_{bbox}$ and $\mathcal{L}^{ti}_{iou}$ are the independent loss of each sample at each timestep. More on computing $\mathcal{L}^{ti}_{bbox}$ and $\mathcal{L}^{ti}_{iou}$ are provided in the supplementary document.

Weighting factor. The 3D localization and orientation estimates at the initial timesteps of the reverse process become more uncertain as the signal-to-noise ratio increases as the reverse process progresses. In order to account for the uncertainty at different signal-to-noise ratio stages, we adopt a weighting factor as follows:

$$\lambda_t = \frac{(1-\beta_t)(1-\bar{\alpha_t})/\beta_t}{(p + \text{SNR}(t))^{\gamma}},$$

where $\text{SNR}(t) = \frac{\bar{\alpha_t}}{1-\bar{\alpha_t}}$ and p and γ are hyperparameters following [12]. With the introduced weighting scheme, we modify the reprojection loss as follows:

$$\mathcal{L}_{\mathit{reproj}} = \sum_{t=1}^{T}\lambda_t\sum_{i=1}^{N} \lambda_{bbox}\mathcal{L}^{ti}_{bbox} + \lambda_{iou}\mathcal{L}^{ti}_{iou}.$$

Training. In the training stage, we take the monocular images and compute the distributions for D and R from the training set. Subsequently, we compute Φ_T as a GMM to initialize the noise sampling procedure. Then, for each sampled ground truth 3D bounding box ϕ^i_0 we generate a set of intermediary samples $\left\{\phi^i_t; t \in \{1,...,T\}\right\}$ using Eq. (1). Next, we pass the monocular image of the object corresponding to the ground truth ϕ^i_0 through the feature encoder Ψ and extract the features ψ^i as the condition to the diffusion process. Finally, the denoising network is optimized using the composite loss function

$$\mathcal{L}_f = \mathcal{L}_{\mathit{diff}} + \lambda_{reproj}\mathcal{L}_{\mathit{reproj}},$$

where λ_{reproj} is a hyperparameter to scale $\mathcal{L}_{reproj}$.

Inference. During testing, we first detect the 2D bounding boxes or the objects of interest using a 2D detector. Then, we initialize ϕ_T for each object of interest and extract the features for the monocular image. We perform the reverse process by recursively feeding to the denoising network to estimate ϕ_0. Once we generate the D and R matrices using the reverse process, we follow the procedure explained in Sec. 3.5 to compute the localization and orientation information in the camera coordinate system.

4. Experiments

4.1. Benchmarks and metrics

KITTI. The KITTI-3D detection [20] benchmark has a *trainval* set and a *test* set with 7,481 images and 7,518 images, respectively. To train MonoDiff for the KITTI dataset, we employ the train-val split used in [9]. Accordingly, we split the *trainval* set as 3,712 training images and 3,769 validation images. To evaluate MonoDiff performance on the KITTI dataset, we use the 3D detection criterion with a 0.7 threshold and report $\text{AP}_{\text{3D(R40)}}$ [74].

Waymo. The Waymo Open Dataset [68] provides 798 *train* and 202 *val* sequences. We adopt the performance reporting criterion of CaDDN [58] for a fair comparison with existing methods. We train MonoDiff on 51,564 samples acquired solely from the front camera. We report the numbers on three ranges at 30m, 50m, and infinity, as well as on two difficulty levels. Performance on the validation set is measured using the official evaluation with 3D IoU criterion at 0.5 threshold. The numbers are reported for mean average

Table 1. AOS and AP_{3D} performance on KITTI *test* set. The **best** and **second-best** figures are in color. The performance metrics for the other methods are reported from the respective published results.

Methods	Venue	AOS				AP_{3D}			
		Average	Easy	Moderate	Hard	Average	Easy	Moderate	Hard
Car									
D4LCN [16]	CVPR'20	78.66	90.01	82.08	63.90	12.63	16.65	11.72	9.51
CaDDN [58]	CVPR'21	68.37	78.28	67.31	59.52	14.68	19.17	13.41	11.46
DDMP-3D [69]	CVPR'21	77.58	90.73	80.20	61.82	14.10	19.71	12.78	9.80
MonoRCNN [63]	ICCV'21	81.70	91.90	86.48	66.71	13.68	18.36	12.65	10.03
MonoJSG [42]	CVPR'22	86.88	92.64	85.00	83.00	18.16	24.69	16.14	13.64
LPCG [51]	ECCV'22	**91.29**	**96.68**	**93.26**	**83.94**	**19.58**	**25.56**	**17.80**	15.38
DID-M3D [52]	ECCV'22	88.45	94.20	90.55	80.61	18.15	24.40	16.29	13.75
MonoNerd [74]	ICCV'23	85.58	94.24	86.13	76.38	18.50	22.75	17.13	**15.63**
MonoDiff		**91.54**	**97.16**	**93.72**	**83.75**	**23.12**	**30.18**	**21.02**	**18.16**
Pedestrian									
D4LCN [16]	CVPR'20	36.35	46.73	33.62	28.71	3.60	4.55	3.42	2.83
CaDDN [58]	CVPR'21	19.12	24.45	17.13	15.79	9.26	12.87	8.14	6.76
DDMP-3D [69]	CVPR'21	35.97	46.12	33.35	28.45	3.83	4.93	3.55	3.01
MonoRCNN [63]	ICCV'21	**43.99**	55.19	**42.59**	34.18	8.11	11.21	7.28	5.85
MonoJSG [42]	CVPR'22	35.82	44.88	32.30	30.27	8.44	11.94	7.36	6.03
LPCG [51]	ECCV'22	43.94	**56.60**	39.79	**35.42**	8.11	10.82	7.33	6.18
DID-M3D [52]	ECCV'22	37.60	46.78	36.37	29.66	8.43	11.78	7.44	6.08
MonoNerd [74]	ICCV'23	22.44	28.43	20.54	18.36	**9.49**	**13.20**	**8.26**	**7.02**
MonoDiff		**46.00**	**58.25**	**43.14**	**36.62**	**9.91**	**13.51**	**8.94**	**7.28**
Cyclist									
D4LCN [16]	CVPR'20	35.57	48.03	31.70	26.99	1.83	2.45	1.67	1.36
CaDDN [58]	CVPR'21	22.56	30.35	19.96	17.38	4.57	7.00	3.41	3.30
DDMP-3D [69]	CVPR'21	33.95	46.42	29.53	25.91	3.00	4.18	2.50	2.32
MonoRCNN [63]	ICCV'21	42.43	54.93	39.89	32.48	2.03	2.89	1.67	1.54
MonoJSG [42]	CVPR'22	38.71	49.31	33.36	33.46	**5.08**	**8.03**	3.87	3.33
LPCG [51]	ECCV'22	**49.20**	**63.07**	**45.24**	**39.28**	4.97	6.98	**4.38**	**3.56**
DID-M3D [52]	ECCV'22	40.63	51.38	38.15	32.35	5.05	7.82	3.95	3.37
MonoNerd [74]	ICCV'23	22.99	30.64	20.13	18.19	3.14	4.79	2.48	2.16
MonoDiff		**52.42**	**67.21**	**48.34**	**41.70**	**5.55**	**8.52**	**4.35**	**3.78**

Table 2. 3D mAP and 3D mAPH pereformance on Waymo *val* set. The **best** and **second-best** figures are in color. The performance metrics for the other methods are reported from the respective published results.

	Methods	Venue	3D mAP				3D mAPH			
			Overall	0 - 30m	30 - 50m	50m - ∞	Overall	0 - 30m	30 - 50m	50m - ∞
LEVEL 1	CaDDN [58]	CVPR'21	17.54	45.00	9.24	0.64	17.31	44.46	9.11	0.62
	MonoJSG [42]	CVPR'22	5.65	20.86	3.91	0.97	5.47	20.26	3.79	0.92
	LPCG [51]	ECCV'22	6.23	18.39	3.44	0.19	6.09	18.03	3.33	0.17
	CMKD [28]	ECCV'22	14.69	38.67	6.26	0.40	14.59	38.44	6.20	0.38
	DID-M3D [52]	ECCV'22	20.66	40.92	15.63	5.35	20.47	40.60	15.48	5.24
	MonoNerd [74]	ICCV'23	**31.18**	**61.11**	**26.08**	**6.60**	**30.70**	**60.28**	**25.71**	**6.47**
	MonoDiff		**32.28**	**63.94**	**25.91**	**7.51**	**31.49**	**62.13**	**25.47**	**7.34**
LEVEL 2	CaDDN [58]	CVPR'21	16.51	44.87	8.99	0.58	16.28	44.33	8.86	0.55
	MonoJSG [42]	CVPR'22	5.34	20.79	3.79	0.85	5.17	20.19	3.67	0.82
	LPCG [51]	ECCV'22	5.84	18.33	3.34	0.17	5.70	17.97	3.23	0.15
	CMKD [28]	ECCV'22	12.99	38.17	5.77	0.38	12.90	37.95	5.71	0.35
	DID-M3D [52]	ECCV'22	19.37	40.77	15.18	4.69	19.19	40.46	15.04	4.59
	MonoNerd [74]	ICCV'23	**29.29**	**60.91**	**25.36**	**5.77**	**28.84**	**60.08**	**25.00**	**5.66**
	MonoDiff		**30.73**	**63.86**	**25.28**	**6.43**	**30.48**	**62.92**	**24.86**	**6.29**

precision and mean average precision weighted by heading annotated as mAP and mAPH, respectively.

4.2. Implementation details

Training details. We implemented MonoDiff using the PyTorch framework and performed the experiments with four NVIDIA A6000 GPUs. We sample four ground truth points per iteration and perform the forward diffusion process for 100 steps. We estimate GMM using five Gaussian components and use DDIM [67] to improve inference speed. The denoising network is trained for 100 epochs with 256 × 256 images. To produce image features, we use the ResNet-34

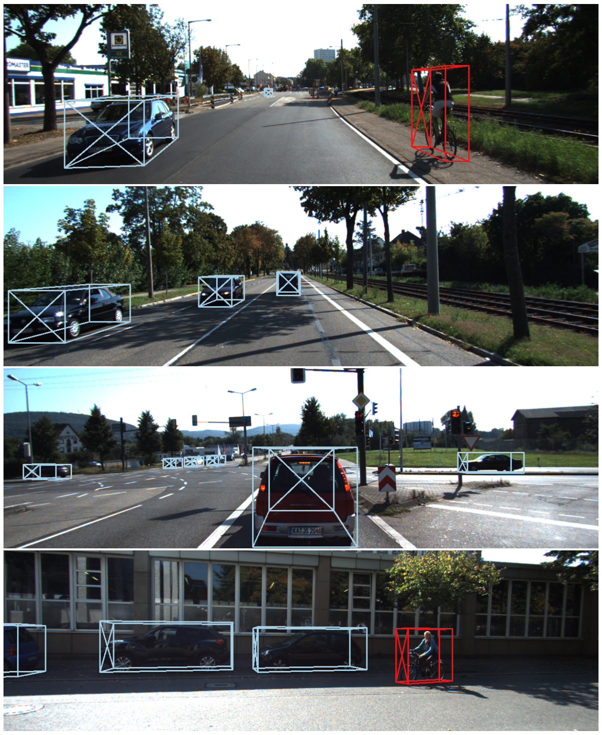

Figure 3. Qualitative results corresponding to MonoDiff.

architecture (pre-trained on ImageNet) as the context encoder Ψ. We use an AdamW optimizer with a fixed learning rate 1e-4 and a linear warm-up schedule over ten epochs.

The 2D detector is trained using AdamW [34] optimizer with a batch size of four and momentum factors 0.9 and 0.999 for β_1 and β_2. On KITTI, we use a 2e-3 learning rate for 75 epochs and then 15 epochs with a 5e-4 learning rate. On Waymo, we train with a learning rate of 1e-3 for the first 20 epochs and a learning rate of 1e-4 for the last ten epochs. In both experiments, we set the weight decay factor to 1e-4.
Hyperparameters. Hyperparameters for the loss function are λ_{bbox}, λ_{iou}, λ_t, and λ_{reproj}. We set λ_{reproj} at 5e-2. Then the parameters γ, and p in λ_t are fixed at 0.5 and 1, respectively following [12]. We use 1 for λ_{bbox} and 0.02 for λ_{iou} for the reprojection loss. Hyperparameters are the same for all experiments. More details are provided in the supplementary document.

4.3. Main results

KITTI *test* set results are tabulated in Tab. 1 where without additional modalities and training data MonoDiff outperforms previous methods. We outperform the most recent SOTA method MonoNeRD [74], which uses stereo images during training. We boost the performance (AOS/AP_{3D}) from 86.13/17.13 to 93.72/21.02 under moderate setting and from 76.38/15.63 to 83.75/18.13 under hard setting for the Car object category, respectively.

Waymo *val* set results are tabulated in Tab. 2 for 3D object detection from the official evaluation. MonoDiff achieves competitive results 32.28/31.49 compared to MonoNeRD 31.18/30.70 on $3D_{mAP}/3D_{mAPH}$ without using additional data.
Qualitative results are presented in Fig. 3 for 3D localization and pose estimation generation with diffusion models.

4.4. Ablation studies

We conduct ablation experiments on KITTI *val* set to validate the impact of each design in our method.
Diffusion process impact. To demonstrate the impact of performing the 3D detection as a generative process, we consider two baselines: (1) **Baseline 1**, which mirrors the architecture of MonoDiff but a single step inference. (2) **Baseline 2**, where the model architecture from Baseline 1 matches the computational complexity of MonoDiff by iterating. The baselines and MonoDiff results are presented in Tab. 3, and the former is inferior to the latter.

Table 3. Ablation study for diffusion pipeline.

Method	AOS	AP_{3D}
	Easy / Moderate / Hard	
Baseline 1	92.90 / 88.75 / 76.76	27.74 / 18.25 / 15.88
Baseline 2	93.11 / 88.23 / 77.96	27.86 / 18.47 / 15.53
MonoDiff	98.46 / 94.72 / 85.75	32.18 / 22.02 / 19.84

MonoDiff component analysis. As tabulated in Tab. 4, we begin with the standard diffusion model and sample noise from the normal distribution. **G**, **F**, **R**, and **W** in Tab. 4 represent GMM initialization, including feature encoder for conditioning, 2D projection supervision, and scaling with the weighting factor, respectively. The standard diffusion model performs better than Baselines 1 and 2, promoting diffusion models (generative models) for discriminative tasks in addition to Tab. 3. According to Tab. 4, initializing with the GMM model of Φ_T improves orientation performance by 3%, while detection performance has improved by $\sim 2\%$. Then, using 2D bounding box information during training has improved the performance of our MonoDiff even though using a 2D detector prohibits end-to-end training. Likewise, using monocular image features instead of the image is conducive to performing the reverse diffusion process even though the feature encoder introduces an extra computational cost. However, since there is only a single

Table 4. Ablation study for different components.

G	F	R	W	AOS	AP_{3D}
				Easy / Moderate / Hard	
✗	✗	✗	✗	94.32 / 89.12 / 78.58	28.44 / 19.99 / 16.52
✓	✗	✗	✗	97.27 / 93.11 / 83.69	31.10 / 21.44 / 18.89
✓	✗	✓	✗	97.64 / 93.61 / 84.33	31.44 / 21.62 / 19.18
✓	✓	✓	✗	98.05 / 94.17 / 85.04	31.81 / 21.82 / 19.51
✓	✓	✓	✓	98.46 / 94.72 / 85.75	32.18 / 22.02 / 19.84

pass through the feature encoder, the additional computation head is negligible compared to the computational cost of the entire reverse process. Finally, adopting the weighting scheme has gained marginal improvements at no expense.

Mixture model ablation. We vary the size of Φ_{GMM} to test the improvement from the GMM and report the numbers in Tab. 5. Furthermore, we discarded the remaining components of our MonoDiff to compare fairly with the standard diffusion forward process. While increasing the number of components improves the performance, it burdens memory and time during training. Moreover, as the size of the GMM increases, the performance tends to saturate at an unnecessary expense. Hence, we chose five Gaussian components to be optimal compute Φ_{GMM}. We do not consider the single Gaussian case as it is equivalent to the standard diffusion model implementation.

Table 5. Ablation study for Φ_{GMM} size

#	AOS	AP_{3D}
	Easy / Moderate / Hard	
3	95.61 / 90.86 / 80.81	29.60 / 20.62 / 17.56
5	97.27 / 93.11 / 83.69	31.10 / 21.44 / 18.89
8	97.46 / 93.37 / 84.02	31.27 / 21.53 / 19.04

Generalization with backbones and detectors. In this part, we conduct experiments on the generalization ability of our MonoDiff using different 2D detectors and feature encoders. For the 2D detectors, we use off-the-shelf detectors, and for feature encoders, we use the popular backbones in object detection for comparison. The running speed and memory are tested on a single NVIDIA RTX A6000 GPU on KITTI val. We compare the performance of our MonoDiff, including running speed, operation, AOS, and AP_{3D}.

Table 6. Ablation study for different detectors and backbones.

		Speed	FLOPS	AOS	AP_{3D}
		(fps)	(G)	Easy / Moderate / Hard	
Detector	YOLOv7	14.1	54.7	98.75 / 94.33 / 85.51	32.06 / 22.12 / 19.67
	FasterRCNN	3.8	35.2	98.26 / 95.18 / 85.94	32.44 / 22.08 / 19.28
	CenterNet	11.7	8.7	98.46 / 94.72 / 85.75	32.18 / 22.02 / 19.84
	RetinaNet	7.3	17.3	97.59 / 95.29 / 85.79	32.35 / 21.95 / 19.73
Backbone	EfficientNet-b3	5.9	9.5	98.07 / 94.67 / 84.40	32.44 / 21.87 / 18.67
	EfficientNet-b5	3.6	23.5	98.14 / 94.75 / 84.32	32.35 / 21.94 / 18.59
	ResNet-34	11.7	3.5	98.46 / 94.72 / 85.75	32.18 / 22.02 / 19.84
	ResNet-50	8.2	4.5	98.15 / 94.97 / 85.88	31.94 / 22.16 / 19.14
	MobileNet	16.2	0.4	96.59 / 94.65 / 84.46	30.73 / 21.85 / 18.72

The results are shown in Tab. 6. According to Tab. 6, the difference in performance for various 2D detectors and feature encoders is not very significant, though the inference speed could vary significantly. Furthermore, in some cases, light models outperform heavy models, especially for the 'Easy' class, while heavy models marginally perform better for the 'Moderate' and 'Hard' classes. Nonetheless, the proposed diffusion pipeline can be adapted to different architectures depending on the practical requirements.

Inference Speed and performance. We tabulate the inference speed (FPS) of state-of-the-art monocular 3D detection methods against the performance on KITTI *test* in Tab. 7 along with MonoDiff results. Moreover, the FPS of our model satisfies most real-time requirements considering other state-of-the-art monocular 3D detection methods. Additionally, we considered the effect of reverse diffusion timesteps and the trade-off between inference speed and performance. The results are illustrated in Fig. 4 for the orientation task. The final number of reverse diffusion steps was selected where the FPS gain is significant with a negligible drop in performance (elbow method).

Table 7. Ablation study for inference speed and performance.

	Speed	FLOPS	AOS	AP_{3D}
	(fps)	(G)	Easy / Moderate / Hard	
LPCG	33.4	16.7	**96.68** / **93.26** / **83.94**	25.56 / 17.80 / 15.38
CMKD	10.1	9.8	95.07 / 89.81 / 83.24	**28.55** / **18.69** / **16.77**
MonoNeRD	12.5	7.2	94.24 / 86.13 / 76.38	22.75 / 17.13 / 15.63
MonoRCNN	14.3	8.5	91.90 / 86.48 / 66.71	18.36 / 12.65 / 10.03
MonoDiff	11.7	14.1	**97.16** / **93.72** / **83.75**	**30.18** / **21.02** / **18.16**

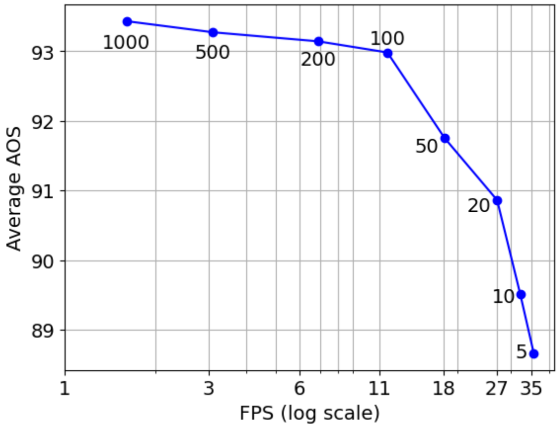

Figure 4. Infernce speed vs AOS performance. Corresponding diffusion steps are shown near the data point.

5. Conclusion

We proposed a novel framework that handles monocular 3D detection and pose estimation as a generative task. MonoDiff allows learning effective representations using diffusion models without additional modalities and training data. Also, by introducing a GMM-based initialization, MonoDiff improved the inference speed and performance of diffusion models for object detection and pose estimation. Furthermore, the MonoDiff pipeline generalizes well to different detectors and backbones while meeting the real-time performance of other state-of-the-art methods.

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-21-2-0211.

References

[1] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 2

[2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021. 2

[3] Markus Braun, Qing Rao, Yikang Wang, and Fabian Flohr. Pose-rcnn: Joint object detection and pose estimation using 3d object proposals. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1546–1551. IEEE, 2016. 2

[4] LE Carvalho and Aldo von Wangenheim. 3d object recognition and classification: a systematic literature review. *Pattern Analysis and Applications*, 22:1243–1292, 2019. 2

[5] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *CVPR*, pages 2040–2049, 2017. 1

[6] 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, 3, 4

[7] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28, 2015. 2

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2156, 2016. 1

[9] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1259–1272, 2017. 2, 5

[10] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12536–12545, 2020. 2

[11] Yi-Nan Chen, Hang Dai, and Yong Ding. Pseudo-stereo for monocular 3d object detection in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 887–897, 2022. 1, 2

[12] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022. 5, 7

[13] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. Monodistill: Learning spatial features for monocular 3d object detection. *arXiv preprint arXiv:2201.10830*, 2022. 2

[14] Adriano D'Alessandro, Ali Mahdavi-Amiri, and Ghassan Hamarneh. Syrac: Synthesize, rank, and count. *arXiv preprint arXiv:2310.01662*, 2023. 3

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[16] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition workshops*, pages 1000–1001, 2020. 6

[17] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1091–1100, 2020. 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] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation from single or multiple model views. *International journal of computer vision*, 67(2):159–188, 2006. 2

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 5

[21] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 3

[22] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang. Diffusioninst: Diffusion model for instance segmentation. *arXiv preprint arXiv:2212.02773*, 2022. 2

[23] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022. 4

[24] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8409–8416, 2019. 1

[25] Jonas Heylen, Mark De Wolf, Bruno Dawagne, Marc Proesmans, Luc Van Gool, Wim Abbeloos, Hazem Abdelkawy, and Daniel Olmeda Reino. Monocinis: Camera independent monocular 3d object detection using instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 923–934, 2021. 1

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 4, 5

[27] Karl Holmquist and Bastian Wandt. Diffpose: Multi-hypothesis human pose estimation using diffusion models.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15977–15987, 2023. 3

[28] Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3d object detection. In *European Conference on Computer Vision*, pages 87–104. Springer, 2022. 1, 6

[29] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020. 1

[30] Siyuan Huang, Yixin Chen, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Perspectivenet: 3d object detection from a single rgb image via perspective points. *Advances in neural information processing systems*, 32, 2019. 2

[31] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 515–532. Springer, 2020. 2

[32] Seong-heum Kim and Youngbae Hwang. A survey on deep learning based methods and datasets for monocular 3d object detection. *Electronics*, 10(4):517, 2021. 1

[33] Se-Ho Kim, Inyong Koo, Inyoung Lee, Byeongjun Park, and Changick Kim. Diffref3d: A diffusion-based proposal refinement framework for 3d object detection. *arXiv preprint arXiv:2310.16349*, 2023. 3

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 7

[35] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009. 2

[36] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 1

[37] Jonathan Li and Andrew Barron. Mixture density estimation. *Advances in neural information processing systems*, 12, 1999. 4

[38] Peixuan Li and Huaici Zhao. Monocular 3d detection with geometric constraint embedding and semi-supervised training. *IEEE Robotics and Automation Letters*, 6(3):5565–5572, 2021. 1

[39] Peiliang Li, Tong Qin, et al. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, 2018. 2

[40] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019. 2

[41] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022. 3

[42] Qing Lian, Peiliang Li, and Xiaozhi Chen. Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1070–1079, 2022. 2, 6

[43] Qing Lian, Botao Ye, Ruijia Xu, Weilong Yao, and Tong Zhang. Exploring geometric consistency for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1685–1694, 2022. 1, 2

[44] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020. 1

[45] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2, 3

[46] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 311–327. Springer, 2020. 2

[47] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019. 1

[48] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2

[49] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 3

[50] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 1

[51] Liang Peng, Fei Liu, Zhengxu Yu, Senbo Yan, Dan Deng, Zheng Yang, Haifeng Liu, and Deng Cai. Lidar point cloud guided monocular 3d object detection. In *European Conference on Computer Vision*, pages 123–139. Springer, 2022. 6

[52] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. Did-m3d: Decoupling instance depth for monocular 3d object detection. In *European Conference on Computer Vision*, pages 71–88. Springer, 2022. 6

[53] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unproject-

ing 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

[54] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5890, 2020. 2

[55] Zequn Qin and Xi Li. Monoground: Detecting monocular 3d objects from the ground. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3793–3802, 2022. 2

[56] Aimon Rahman, Jeya Maria Jose Valanarasu, Ilker Hacihaliloglu, and Vishal M Patel. Ambiguous medical image segmentation using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11536–11546, 2023. 3

[57] Yasiru Ranasinghe, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M Patel. Diffuse-denoise-count: Accurate crowd-counting with diffusion models. *arXiv preprint arXiv:2303.12790*, 2023. 2, 3

[58] 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. 5, 6

[59] Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Image deblurring with domain generalizable diffusion models. *arXiv preprint arXiv:2212.01789*, 2022. 2

[60] Cédric Rommel, Eduardo Valle, Mickaël Chen, Souhaiel Khalfaoui, Renaud Marlet, Matthieu Cord, and Patrick Pérez. Diffhpe: Robust, coherent 3d human pose lifting with diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3220–3229, 2023. 2

[61] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International journal of computer vision*, 66: 231–259, 2006. 2

[62] 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. 2

[63] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15172–15181, 2021. 6

[64] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 1

[65] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Peter Kontschieder, and Elisa Ricci. Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3225–3233, 2021. 1

[66] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2, 3

[67] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3, 5, 6

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 5

[69] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 454–463, 2021. 6

[70] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 2

[71] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2

[72] Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6d-vnet: End-to-end 6-dof vehicle pose estimation from monocular rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[73] Junde Wu, Rao Fu, Huihui Fang, Yu Zhang, Yehui Yang, Haoyi Xiong, Huiying Liu, and Yanwu Xu. Medsegdiff: Medical image segmentation with diffusion probabilistic model. *arXiv preprint arXiv:2211.00611*, 2022. 3

[74] Junkai Xu, Liang Peng, Haoran Cheng, Hao Li, Wei Qian, Ke Li, Wenxiao Wang, and Deng Cai. Mononerd: Nerf-like representations for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6814–6824, 2023. 1, 2, 3, 5, 6, 7

[75] Zhenbo Xu, Wei Zhang, Xiaoqing Ye, Xiao Tan, Wei Yang, Shilei Wen, Errui Ding, Ajin Meng, and Liusheng Huang. Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12557–12564, 2020. 2

[76] Jieming Zhou, Tong Zhang, Zeeshan Hayder, Lars Petersson, and Mehrtash Harandi. Diff3dhpe: A diffusion model for 3d

human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2092–2102, 2023. 3

[77] Zheyuan Zhou, Liang Du, Xiaoqing Ye, Zhikang Zou, Xiao Tan, Li Zhang, Xiangyang Xue, and Jianfeng Feng. Sgm3d: Stereo guided monocular 3d object detection. *IEEE Robotics and Automation Letters*, 7(4):10478–10485, 2022. 1

[78] Minghan Zhu, Songan Zhang, Yuanxin Zhong, Pingping Lu, Huei Peng, and John Lenneman. Monocular 3d vehicle detection using uncalibrated traffic cameras through homography. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3814–3821. IEEE, 2021. 1

[79] 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. 4

[80] Zhikang Zou, Xiaoqing Ye, Liang Du, Xianhui Cheng, Xiao Tan, Li Zhang, Jianfeng Feng, Xiangyang Xue, and Errui Ding. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2713–2722, 2021. 2