PRIORDIFFUDINO: SELF-TRAJECTORY DENOISING FOR DIFFUSION-BASED DETECTION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We construct the diffusion process of a detection transformer as a denoising trajectory that starts from the noised encoder proposals and extends along the model's own predictions. We call this self-trajectory denoising and instantiate it on DINO as PriorDiffuDINO. The starting point of the trajectory does not depend on the ground truth and is shared by training and inference. Because the decoder is also trained on re-noised copies of its own predictions, which are the inputs of the later inference steps, the exposure bias between training and inference is reduced. We further adopt the matchability-aware loss and dense augmentation of DEIM, which also apply to diffusion-based detectors. With a ResNet-50 backbone and 12 training epochs, PriorDiffuDINO reaches 51.3 AP on the COCO 2017 validation set, 2.7 AP above DiffuDINO under the same schedule and close to DiffuDINO trained for 50 epochs (51.9 AP). With a ResNet-101 backbone it reaches 52.4 AP. On LVIS with ResNet-50, it substantially improves every metric over DINO and DiffuDINO, by 3.0 AP overall and 5.3 AP on the rare categories over DiffuDINO.

1 INTRODUCTION

A detector has to localize every object in an image and assign it a category label. Detection transformers cast this as set prediction: a fixed set of queries is decoded into boxes and labels and matched one-to-one with the ground truth, with no anchors and no NMS. Diffusion-based detectors go one step further and treat the boxes, or the reference points of the queries, as a variable that is noised in training and denoised at inference. How this denoising process should be constructed is the subject of this paper.

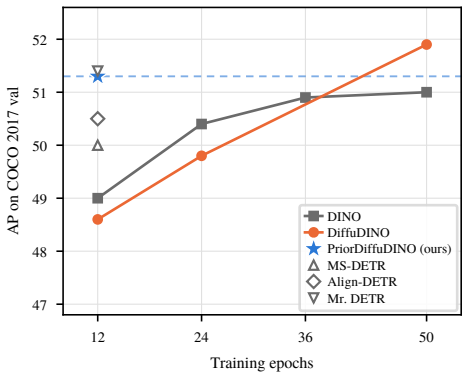

Figure 1: Comparison of PriorDiffuDINO with other methods on COCO 2017 using a ResNet-50 backbone.

Early detectors score and refine a large set of candidate regions, generated from dense anchors in two-stage (Ren et al., 2015) and one-stage (Lin et al., 2017) methods or from every feature location in anchor-free ones (Tian et al., 2019). They are accurate and well studied, but they depend on hand-designed components such as anchor generation, heuristic label assignment (Zhang et al., 2020) and non-maximum suppression (NMS), each of which has to be tuned to the data. DETR (Carion et al., 2020) removes these components by solving set prediction directly with a transformer (Vaswani et al., 2017) and bipartite matching. The price is slow and unstable training, and DETR is trained for 500 epochs on COCO.

Later work makes the queries more informative and the training more stable. Deformable DETR (Zhu et al., 2021) attends to a few sampling points around a reference point of each query, and DAB-DETR (Liu et al., 2022) formulates the queries as anchor boxes. Both give the queries a spatial prior and speed up convergence. DN-DETR (Li et al., 2022) stabilizes the matching with query denoising. DINO (Zhang et al., 2023) combines contrastive denoising with query positions initialized from the encoder proposals and reaches strong accuracy in 12 epochs. More recent work

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

adds denser supervision (Zong et al., 2023) and aligns the classification score with localization quality (Huang et al., 2025). At inference, however, these detectors run the decoder once over a fixed set of queries.

Diffusion models (Song et al., 2021a) have been used to turn detection into an iterative denoising process. DiffusionDet (Chen et al., 2023b) learns to recover ground-truth boxes from noised copies and, at inference, refines boxes sampled from a Gaussian distribution over several steps. The number of boxes and steps can then be chosen at test time. DiffuDETR (Nawar et al., 2026) brings this formulation to detection transformers by diffusing the reference points of the queries, and its DiffuDINO variant reports 51.9 AP on COCO, above the 50.9 AP of DINO. The gain comes at a cost. DiffuDINO needs 50 training epochs to surpass DINO, and its accuracy peaks at three denoising steps and then declines. Running more steps therefore does not help.

The decoder of such a detector is a conditional denoiser. In DiffuDINO every training input is a noised copy of the ground-truth set. At inference the ground truth is unavailable. The first step starts from Gaussian noise, which carries no information about the image, and every later step denoises reference points computed from the model's own predictions, an input the decoder has not seen in training. This is the exposure bias known from sequence models (Bengio et al., 2015) and also observed in diffusion models (Ning et al., 2023). A diffusion-based detector should therefore start from a point that is available without ground truth, identically in training and inference, and be trained on the inputs that its own inference produces.

We propose self-trajectory denoising and instantiate it on DINO as PriorDiffuDINO. The denoising trajectory starts from the proposals of the encoder, which are available without ground truth in both training and inference, and the decoder is also trained along its own predictions. The later inference steps thus receive inputs of the kind seen in training. We further find that training practices of recent DETR-based detectors (Huang et al., 2025) carry over to this generative setting. With a ResNet-50 backbone (He et al., 2016) and 12 training epochs, PriorDiffuDINO reaches 51.3 AP on COCO 2017 (Lin et al., 2014), close to DiffuDINO trained for 50 epochs (Figure 1), and its accuracy varies by at most 0.2 AP from one to ten denoising steps.

Our contributions are summarized as follows:

- We point out the exposure bias of diffusion-based detection transformers and propose self-trajectory denoising. The denoising trajectory starts from the encoder proposals in both training and inference, and the decoder is trained along its own predictions.
- We introduce two training practices of DETR-based detectors into diffusion-based detection transformers. The matchability-aware loss ties the classification score to localization quality, and dense augmentation raises the number of positive queries per image.
- Trained for only 12 epochs, PriorDiffuDINO improves over both DINO and DiffuDINO on COCO 2017 and on LVIS. It reaches 51.3 AP on COCO ($+2.3$ over DINO and $+2.7$ over DiffuDINO) and 31.9 AP on LVIS ($+5.4$ and $+3.0$ respectively), with the largest gain on rare categories.

2 Related Work

2.1 Detection Transformers

DETR (Carion et al., 2020) decodes a fixed set of object queries with a transformer (Vaswani et al., 2017) and trains them by bipartite matching, without anchors or NMS, but converges slowly. The follow-up work below concerns the starting point of the queries, denoising training and the training objective.

Query initialization Deformable DETR (Zhu et al., 2021) lets each query attend to a few sampling points around a reference point, and its two-stage variant takes the reference points from the top-scoring proposals of the encoder, as does Efficient DETR (Yao et al., 2021). Conditional DETR (Meng et al., 2021), SMCA (Gao et al., 2021), DAB-DETR (Liu et al., 2022) and Anchor DETR (Wang et al., 2022) give each query an explicit spatial prior, and DAB-DETR also updates it layer by layer. In these detectors the queries are decoded once from their initial reference points, without noise. We also start from the encoder proposals, but use them as the noisy prior of a denoising trajectory.

Denoising training DN-DETR (Li et al., 2022) feeds noised ground-truth boxes and labels to the decoder as auxiliary groups of queries with a fixed assignment, which stabilizes the matching. DINO (Zhang et al., 2023) extends this to contrastive denoising and initializes the positions of the queries from the encoder proposals. The noised queries of both methods are built from the ground truth and exist only in training. We keep the contrastive denoising group and also noise the queries that produce the detections, in training and at inference.

Training objectives Co-DETR (Zong et al., 2023) attaches auxiliary heads trained with one-to-many label assignment, and H-DETR (Jia et al., 2023) and Group DETR (Chen et al., 2023a) add hybrid or group-wise matching branches. Another line aligns the classification objective with localization quality, an idea also explored for dense detectors (Feng et al., 2021). Stable-DINO (Liu et al., 2023) supervises the classification scores of the matched queries with their IoU and modulates the matching cost by localization quality. Align-DETR (Cai et al., 2024) and Rank-DETR (Pu et al., 2023) introduce IoU-aware and rank-oriented losses. DEIM (Huang et al., 2025), built on RT-DETR (Zhao et al., 2024b) and D-FINE (Peng et al., 2025), combines a matchability-aware loss with dense augmentation by mosaic (Bochkovskiy et al., 2020) and mixup (Zhang et al., 2018).

2.2 Diffusion-Based Detection

Pix2seq (Chen et al., 2022) treats detection as generation and outputs box coordinates and class labels as discrete tokens. Diffusion models (Song et al., 2021b) learn to reverse a gradual noising process, and DDIM (Song et al., 2021a) samples from them in a few deterministic steps.

DiffusionDet (Chen et al., 2023b) diffuses the box coordinates and refines them with a Sparse R-CNN (Sun et al., 2021) decoder. At inference it starts from Gaussian boxes and replaces low-scoring boxes by new samples between steps, which the authors call box renewal. The number of boxes and steps can be chosen at test time with the same weights.

DiffuDETR (Nawar et al., 2026) brings this formulation to detection transformers. It generates the reference points of the queries from a Gaussian prior, with DiffuDETR on Deformable DETR and DiffuDINO on DINO, and its sampling reruns only the decoder at each step. Both variants train the decoder on noised copies of the ground truth and start inference from Gaussian noise (Section 3.1). Our decoder starts from the encoder proposals, is trained on its own predictions and needs neither box renewal nor NMS across steps.

2.3 Exposure Bias and Self-Conditioning

A model trained on ground-truth inputs but run on its own outputs suffers from exposure bias. Scheduled sampling (Bengio et al., 2015) counters it in sequence prediction by gradually replacing the true previous token with the generated one during training. Diffusion models show the same mismatch. Input perturbation (Ning et al., 2023) perturbs the training inputs to simulate the errors of inference, and later work corrects the mismatch at sampling time without retraining, by scaling down the network output (Ning et al., 2024) or by shifting the timesteps (Li et al., 2024). Our training differs from input perturbation in the source of the errors. It uses the actual errors of the model's own predictions instead of simulated ones.

Self-conditioning (Chen et al., 2023c) feeds the previous estimate back as an additional input and obtains it in training from a first forward pass with the gradient stopped. We build on this two-pass training. The re-noised first-pass prediction becomes the input of the second pass, which trains the decoder on the inputs of its own inference.

3 Method

3.1 Preliminaries

Reference points in DINO DINO (Zhang et al., 2023) decodes a set of queries, each with a content part and a reference point, a box that locates the image features the query attends to and is refined layer by layer. The reference points of the main queries come from the top-k encoder proposals, and a contrastive denoising group of noised ground-truth boxes is added during training.

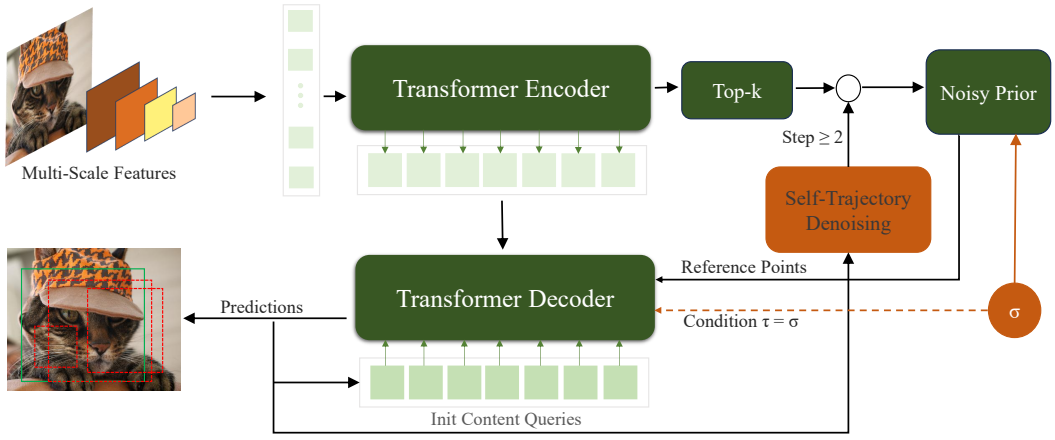

Figure 2: Overview of PriorDiffuDINO. The top-k proposals of the encoder are perturbed to form the noisy prior, from which the decoder starts with the noise level σ as its condition. Self-trajectory denoising feeds the predictions back to the prior, and the decoder denoises along its own predictions. Training runs the decoder twice and inference runs it k times.

Diffusion over reference points DiffuDINO (Nawar et al., 2026) makes these reference points the variable of a diffusion process. In training, the ground-truth boxes r_0, padded with random boxes, are noised by the forward process of DDPM (Ho et al., 2020), $r_t = \sqrt{\bar{\alpha}_t}\, r_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon$, and the decoder, conditioned on t, learns to recover the boxes. At inference the reference points start from Gaussian noise and are denoised by DDIM (Song et al., 2021a).

Exposure bias The cosine schedule drives $\bar{\alpha}_t$ to zero at the last timestep. The Gaussian starting point of inference is therefore the endpoint of the forward process, and the first step is covered by training. Every later step receives reference points computed from the model's own estimate, whereas every training input is derived from the ground truth. The released implementation handles this exposure bias (Section 2.3) with sampling heuristics, namely box renewal between steps and NMS over the pooled predictions of all steps.

3.2 Self-Trajectory Denoising

PriorDiffuDINO is built on the architecture of DINO and changes how the reference points of the main queries are formed and trained (Figure 2).

Denoising trajectory Let b be the top-k proposals of the encoder with coordinates in $(0, 1)$, q the learned content queries and D the decoder, whose image features and contrastive denoising group we omit from the notation. We describe detection as a trajectory of boxes $\hat{b}^{(i)}$ and query states $h^{(i)}$ that starts from $\hat{b}^{(0)} = b$ and $h^{(0)} = q$. Step i perturbs the current boxes and decodes them,

$$r^{(i)} = \mathrm{sigmoid}\big(\mathrm{logit}(\hat{b}^{(i-1)}) + \sigma_i\, \epsilon_i\big), \qquad \epsilon_i \sim \mathcal{N}(0, I), \tag{1}$$

$$\big(\hat{b}^{(i)}, h^{(i)}\big) = D\big(r^{(i)},\, h^{(i-1)};\, \tau = \sigma_i\big), \tag{2}$$

where $r^{(i)}$ are the reference points of the main queries, σ_i is the noise level of the step, and $\hat{b}^{(i)}$ and $h^{(i)}$ are the boxes and the query states of the last decoder layer. The first step perturbs the encoder proposals. Every later step perturbs the model's own previous prediction and carries its query states as the content of the queries. Training and inference apply the same step and differ only in the number of steps and in how the noise levels are chosen. Unlike the forward process of DDPM, which destroys the signal at the last timestep, the perturbation keeps the reference points around the current boxes, since $\sigma_i \le \sigma_{\max} = 0.5$ in logit space. Each step is thus a local correction of the current boxes, not a generation from noise.

Noisy prior from encoder proposals The encoder proposals are computed from the image alone and are therefore available in training and at inference alike. They also place the reference points

where the encoder expects objects, while Gaussian samples carry no information about the image. Removing this prior at inference costs 7.7 AP with one step, mostly on small objects (Section 4.3). The noise is added in logit space. There a perturbation cannot move a box out of the valid range, and for a small width w it acts multiplicatively, $\mathrm{sigmoid}(\mathrm{logit}(w) + \sigma\epsilon) \approx w\, e^{\sigma\epsilon}$, so a small box receives a proportionally small perturbation. The proposals act as a noisy prior of the trajectory, and small objects are not swamped by the noise. The encoder is supervised on the clean proposals. The proposal head therefore learns to produce the prior and does not compensate for its noise.

Noise-level conditioning The timestep of each step is its noise level, $\tau = \sigma_i$. It is injected by FiLM. Every decoder layer maps an embedding of τ to per-channel coefficients $s(\tau)$ and $u(\tau)$ and transforms a query feature x into

$$\tilde{x} = s(\tau) \odot x + u(\tau). \tag{3}$$

The condition tells the decoder how strongly its reference points are perturbed, and one set of weights covers every noise level of training and inference. The contrastive denoising group is conditioned on $\tau = 1$.

Training along its own trajectory Training runs two steps with $\sigma_1, \sigma_2 \sim U(0, \sigma_{\max})$ and $\sigma_{\max} = 0.5$ (Figure 3). The second step takes a re-noised copy of the model's own prediction, the input of every later inference step, and the decoder is trained on the inputs that its inference produces. This reduces the exposure bias without sampling heuristics. It extends self-conditioning (Chen et al., 2023c), which feeds the previous estimate as an extra input, by making the re-noised estimate the input itself. The gradient is stopped between the two steps. At inference the previous prediction is a fixed input, and the stopped gradient treats it the same way in training while avoiding backpropagation through two decoder passes. The query states are carried together with the boxes, so the content of each query keeps describing the box it refines.

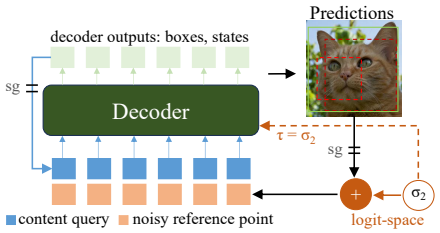

Figure 3: Self-trajectory denoising. The predicted boxes are perturbed again to form the noisy prior of the next pass, and the decoder outputs are carried over as the content queries.

Inference along one trajectory Inference runs k steps with $\sigma_i = \sigma_{\max}(1 - \frac{i-1}{k})$ and outputs the last one. The larger noise of the early steps is meant to let the reference points leave poorly placed proposals, and the smaller noise of the later steps to preserve the refined boxes. The queries are trained with one-to-one matching and stay on one trajectory. The last step thus contains no duplicates across steps, and box renewal, merging of steps and NMS are unnecessary. The accuracy stays within 0.2 AP from one to ten steps (Section 4.3).

3.3 Incorporating Advances in DETR Training

Every denoising step still produces its outputs by set prediction, with bipartite matching, a classification loss and the L1 and GIoU (Rezatofighi et al., 2019) box losses. Improvements of DETR training therefore carry over to every step, and we adopt two practices of DEIM (Huang et al., 2025).

Matchability-aware loss The predictions are ranked by their scores, so a high score on a poorly placed box lowers the AP. Let p be the predicted probability of a query for a class, q the IoU of its box with the assigned ground truth, and $y = 1$ if the query is matched to an object of this class and $y = 0$ otherwise. The classification loss is

$$\mathcal{L}_{\mathrm{MAL}}(p, q, y) = \begin{cases} -q^{\gamma} \log p - (1 - q^{\gamma}) \log(1 - p), & y = 1, \\ -p^{\gamma} \log(1 - p), & y = 0, \end{cases} \tag{4}$$

with $\gamma = 1.5$. A matched query is trained toward q^{γ} and an unmatched one is penalized in proportion to its confidence, which ties the score to the localization quality. This is useful here because every step starts from perturbed reference points, and the score should reflect how well the resulting box is placed.

Table 1: Comparison of object detectors on the COCO 2017 validation set using ResNet-50 and ResNet-101 backbones.

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 (He et al., 2016)							
DETR-DC5 (Carion et al., 2020)	500	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR (Zhu et al., 2021)	50	46.2	65.2	50.0	28.8	49.2	61.7
DN-Deformable DETR (Li et al., 2022)	50	48.6	67.4	52.7	31.0	52.0	63.7
DINO (Zhang et al., 2023)	36	50.9	69.0	55.3	34.6	54.1	64.6
MS-DETR (Zhao et al., 2024a)	24	50.9	68.4	56.1	34.7	54.3	65.1
Salience DETR (Hou et al., 2024)	24	51.2	68.9	55.7	33.9	55.5	65.6
Align-DETR (Cai et al., 2024)	24	51.7	69.0	56.3	35.5	55.0	66.1
Mr. DETR (Zhang et al., 2025)	24	52.3	69.5	56.7	35.2	56.0	67.0
DINO (Zhang et al., 2023)	12	49.0	66.6	53.5	32.0	52.3	63.0
Salience DETR (Hou et al., 2024)	12	49.2	67.1	53.8	32.7	53.0	63.1
MS-DETR (Zhao et al., 2024a)	12	50.0	67.3	54.4	31.6	53.2	64.0
Align-DETR (Cai et al., 2024)	12	50.5	67.7	55.3	34.7	53.6	64.6
Mr. DETR (Zhang et al., 2025)	12	51.4	68.6	55.7	33.8	54.7	66.3
Pix2seq-DC5 (Chen et al., 2022)	300	43.2	61.0	46.1	26.6	47.0	58.6
DiffusionDet (Chen et al., 2023b)	–	46.8	65.3	51.8	29.6	49.3	62.2
DiffuDETR (Nawar et al., 2026)	50	50.2	66.8	55.2	33.3	53.9	65.8
DiffuAlignDETR (Nawar et al., 2026)	24	51.9	69.2	56.4	34.9	55.6	66.2
DiffuDINO (Nawar et al., 2026)	50	51.9	69.4	55.7	35.8	55.7	67.1
DiffuDINO	12	48.6	65.5	52.9	30.7	52.4	64.1
PriorDiffuDINO (ours)	12	51.3	68.4	56.1	35.2	55.0	66.9
ResNet-101 (He et al., 2016)							
DETR-DC5 (Carion et al., 2020)	500	44.9	64.7	47.7	23.7	49.5	62.3
DAB-DETR-DC5 (Liu et al., 2022)	50	46.6	67.0	50.2	28.1	50.5	64.1
DN-DETR-DC5 (Li et al., 2022)	50	47.3	67.5	50.8	28.6	51.5	65.0
DINO (Zhang et al., 2023)	12	50.0	67.7	54.4	32.2	53.4	64.3
Pix2seq-DC5 (Chen et al., 2022)	300	45.0	63.2	48.6	28.2	48.9	60.4
DiffusionDet (Chen et al., 2023b)	–	47.5	65.7	52.0	30.8	50.4	63.1
DiffuDINO (Nawar et al., 2026)	12	51.2	68.6	55.8	33.2	55.6	67.2
DiffuAlignDETR (Nawar et al., 2026)	12	51.7	69.3	56.1	34.0	55.6	67.0
PriorDiffuDINO (ours)	12	**52.4**	**69.9**	**56.9**	**36.4**	**55.8**	**67.5**

Dense augmentation Under one-to-one matching an image provides as many positive queries as it has objects. Dense augmentation composes a training image from several images by mosaic (Bochkovskiy et al., 2020) and mixup (Zhang et al., 2018) during part of the schedule, which raises the number of objects and thus of positive queries per image.

4 Experiments

4.1 Setup

Datasets and metrics We use COCO 2017 (Lin et al., 2014) and LVIS v1 (Gupta et al., 2019). COCO contains 80 object categories, and LVIS contains 1203 categories with a long-tailed distribution. We train on the training split of each dataset and evaluate on its validation split. The main metric is the average precision (AP) over ten IoU thresholds from 0.5 to 0.95. AP_{50} and AP_{75} are the AP at IoU thresholds of 0.5 and 0.75. AP_S, AP_M and AP_L are the AP of small, medium and large objects, whose areas are below 32^2, between 32^2 and 96^2, and above 96^2 pixels. On LVIS, AP_r, AP_c and AP_f are the AP of the rare, common and frequent categories, which appear in 1–10, 11–100 and more than 100 training images.

Baselines We compare with two groups of detectors. The first consists of DETR-based detectors: DETR, Deformable DETR, DAB-DETR, DN-DETR, DN-Deformable DETR, DINO, MS-DETR,

Table 2: Comparison of object detectors on the LVIS validation set using ResNet-50 and ResNet-101 backbones. All models are trained for 12 epochs.

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP_r	AP_c	AP_f
ResNet-50 (He et al., 2016)									
DINO (Zhang et al., 2023)	26.5	35.9	27.8	20.0	35.2	40.9	9.2	24.6	36.2
DiffuDINO (Nawar et al., 2026)	28.9	38.5	30.8	20.7	37.5	46.4	13.7	27.6	36.9
PriorDiffuDINO (ours)	**31.9**	**43.0**	**34.2**	**23.4**	**41.2**	**48.1**	**19.0**	**30.8**	**38.7**
ResNet-101 (He et al., 2016)									
DINO (Zhang et al., 2023)	30.9	40.4	32.8	23.2	40.5	46.3	13.9	29.7	39.7
DiffuDINO (Nawar et al., 2026)	32.5	42.4	34.8	23.5	43.4	49.7	13.5	32.0	**41.5**
PriorDiffuDINO (ours)	**34.5**	**45.5**	**37.0**	**25.8**	**43.9**	**50.8**	**24.1**	**32.5**	41.3

Salience DETR, Align-DETR and Mr. DETR. The second consists of generative detectors: Pix2seq, DiffusionDet, DiffuDETR, DiffuAlignDETR and DiffuDINO.

Configuration We use ResNet-50 and ResNet-101 backbones (He et al., 2016) initialized from ImageNet, four feature scales and a schedule of 12 epochs. Appendix A gives all implementation details and hyper-parameters, and the code will be released after the double-blind review.

4.2 Main Results

Table 1 compares PriorDiffuDINO with existing methods on COCO. With a ResNet-50 backbone and 12 training epochs, PriorDiffuDINO reaches 51.3 AP. It surpasses most DETR-based detectors in the table, including DINO trained for 36 epochs and MS-DETR and Salience DETR trained for 24 epochs. Among the 12-epoch results it is level with Mr. DETR and 2.3 AP above DINO. The most optimized variants remain ahead at 24 epochs, Align-DETR by 0.4 AP and Mr. DETR by 1.0 AP, and PriorDiffuDINO narrows this gap with half of their training schedule. Among the generative detectors, PriorDiffuDINO is above Pix2seq, DiffusionDet and DiffuDETR trained for 50 epochs. Under the same 12-epoch schedule it is 2.7 AP above DiffuDINO, and it comes within 0.6 AP of DiffuAlignDETR and DiffuDINO trained for 24 and 50 epochs, with half and a quarter of their schedules. With a ResNet-101 backbone, PriorDiffuDINO leads on every metric, 0.7 AP above DiffuAlignDETR and 2.4 AP above DINO. The gains over both DINO and DiffuDINO at the same schedule show that our construction of the denoising process and the adopted training practices are effective. Section 4.3 examines each of them.

Table 2 reports the comparison on LVIS, where each category is supported by far fewer images than on COCO and the supervision of the rare categories is sparse. PriorDiffuDINO reaches 31.9 AP with ResNet-50 and 34.5 with ResNet-101, $+3.0$ and $+2.0$ over DiffuDINO under the same schedule. The margin over DINO is $+5.4$ and $+3.6$. With ResNet-50, PriorDiffuDINO even exceeds DINO with ResNet-101 on every metric except AP_f. It is also above DiffuDINO with ResNet-101 in AP_{50} and AP_r, and within 0.6 AP of it in AP, AP_{75} and AP_S. This comparison across backbones highlights the advantage of our method. The improvement is not uniform across the vocabulary. On the frequent categories the three models lie within 2.5 AP of each other. On the rare categories PriorDiffuDINO reaches 19.0 AP_r with ResNet-50 against 13.7 for DiffuDINO, and 24.1 with ResNet-101 against 13.5, gains of $+5.3$ and $+10.6$. DiffuDINO improves AP_r over DINO with ResNet-50 but not with ResNet-101 (13.5 against 13.9). Our model improves it with both backbones.

Figure 4 compares detections on images of the COCO 2017 validation set with crowded small objects, overlapping objects and objects of widely different scales. PriorDiffuDINO produces fewer missed and duplicate detections than DINO, and its results are close to those of DiffuDINO.

4.3 Ablation

Contribution of each component Table 3 ablates the full model, changing one component at a time. The first ablation (w/o Prior) concerns inference only. The decoder starts from boxes sampled

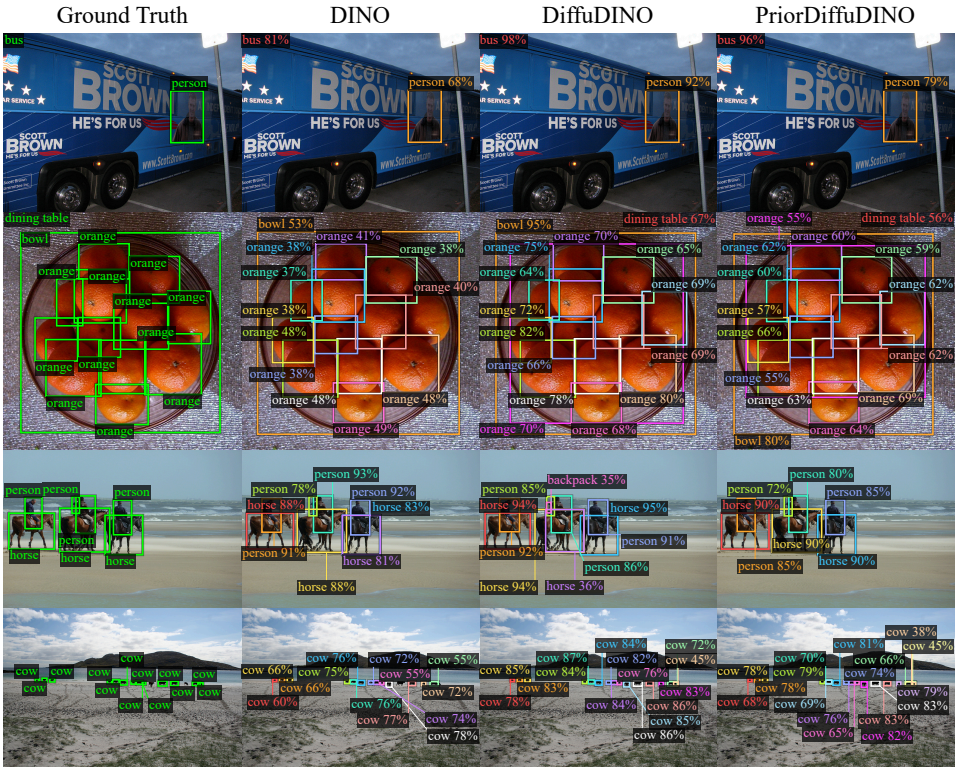

Figure 4: Qualitative comparison on the COCO 2017 validation set with a ResNet-50 backbone.

Table 3: Ablation of PriorDiffuDINO on the COCO 2017 validation set with a ResNet-50 backbone. Prior: the noisy prior from the encoder proposals, replaced by Gaussian boxes at inference only. DA: dense augmentation. MAL: matchability-aware loss, replaced by the focal loss.

Configuration	Steps	AP	AP_S	AP_L
PriorDiffuDINO	1	51.2	35.0	66.4
	3	51.3	35.2	66.9
w/o Prior	1	43.5	23.1	66.1
	3	50.5	33.5	66.7
w/o DA	1	49.7	32.4	64.9
	3	49.7	31.8	65.0
w/o MAL	1	50.6	35.0	64.6
	3	50.7	35.2	65.2

from a Gaussian distribution, as in DiffuDINO, in place of the noised encoder proposals, and the sampler is unchanged. With one step the AP drops by 7.7, from 51.2 to 43.5. In training, the main group receives small perturbations of boxes produced by the model itself, the encoder proposals in the first step and its own predictions in the second. Gaussian boxes are neither, and the first step has to denoise an input far outside this range. From the second step on, the input is a re-noised copy of the model's own prediction, the input form of the second training step, and the accuracy recovers quickly. With three steps the Gaussian start reaches 50.5, within one AP of the full model. The drop is concentrated on the small objects, whose AP falls from 35.0 to 23.1 with one step and is still 1.7 below with three. The large objects are barely affected (66.1 against 66.4, and 66.7 against 66.9). The encoder prior thus matters most for the small objects. The other two rows change the training recipe. Without the dense augmentation (w/o DA) the AP drops by 1.5 and does not recover with more steps. This gain comes from training and cannot be recovered at inference. With the focal loss in place of the matchability-aware loss (w/o MAL) the AP drops by 0.6 with one step and with three.

Table 4: Number of denoising steps: AP on the COCO 2017 validation set with one to ten steps.

Model	Backbone	Epochs	1	3	5	10
DiffuDINO (Nawar et al., 2026)	ResNet-50	50	51.6	51.9	51.8	51.4
PriorDiffuDINO	ResNet-50	12	51.2	51.3	51.4	51.3
PriorDiffuDINO	ResNet-101	12	52.4	52.4	52.4	52.3

The drop comes from the large objects, whose AP falls by 1.8 and 1.7, while the small objects keep the same AP.

Number of denoising steps A diffusion-based detector can trade computation for accuracy by running the decoder for several denoising steps. DiffuDINO reaches 51.6 AP with one step, 51.9 with three, 51.8 with five and 51.4 with ten. Three decoder passes buy 0.3 AP over one, and further steps lose accuracy. Table 4 reports the same measurement for PriorDiffuDINO. With ResNet-50 one step gives 51.2 AP and more steps reach 51.4, and with ResNet-101 the accuracy stays between 52.3 and 52.4 AP. Across one to ten steps it varies by at most 0.2 AP. The two parts of our design play different roles here. The encoder prior makes the first step already accurate. Training along the model's own trajectory prepares the decoder for the inputs of the later steps, and the accuracy remains stable as more steps are run. The w/o Prior row of Table 3 shows this second role directly, as the later steps recover the accuracy from 43.5 to 50.5 AP within three steps. The number of steps thus becomes a free choice. The numbers of PriorDiffuDINO are the output of the last step of a single trajectory, without box renewal, merging of steps or NMS.

5 Conclusion

We present PriorDiffuDINO, which constructs the diffusion process of a detection transformer as self-trajectory denoising. The denoising trajectory starts from the noised proposals of the encoder, a prior available without ground truth in training and at inference, and the decoder is also trained on re-noised copies of its own predictions. These changes address the exposure bias of diffusion-based detection transformers, whose decoder is trained only on inputs derived from the ground truth but denoises its own predictions at inference. We further introduce effective training practices of DETR-based detectors. Trained for only 12 epochs, PriorDiffuDINO reaches 51.3 AP on COCO with ResNet-50. It is 2.7 AP above DiffuDINO under the same schedule, close to DiffuDINO trained for 50 epochs and level with the strongest 12-epoch DETR-based detector. On LVIS with ResNet-50, it improves every metric over DINO and DiffuDINO, by 3.0 AP overall and 5.3 AP on the rare categories over DiffuDINO. With this lighter backbone it also exceeds DINO with ResNet-101 on all metrics except AP_f and comes within 0.6 AP of DiffuDINO with ResNet-101. The ablation shows that the encoder prior, dense augmentation and the matchability-aware loss each contribute, and the accuracy stays within 0.2 AP from one to ten denoising steps. By reconsidering how the denoising process is constructed, PriorDiffuDINO combines diffusion models with DINO more closely and makes the combination more competitive.

AI use statement

During the writing of this paper, we used a large language model to check the grammar, sentence errors and accuracy of wording of our text. The aim was a text that follows academic writing conventions and expresses our meaning clearly. The model was used only on the wording. The research ideas, the method, the experiments and their analysis are the authors' own work. Every AI-assisted edit was reviewed by the authors, and we take responsibility for the entire content of this paper.

Reproducibility statement

Appendix A gives the architecture and the implementation details of our model, including all hyper-parameters (Table 5) and the training and inference procedures (Algorithms 1 and 2). The code will be released after the double-blind review.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Zhi Cai, Songtao Liu, Guodong Wang, Zeming Li, Zheng Ge, Xiangyu Zhang, and Di Huang. Align-DETR: Enhancing end-to-end object detection with aligned loss. In *British Machine Vision Conference*, 2024.

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*, 2020.

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 *International Conference on Computer Vision*, 2023a.

Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. DiffusionDet: Diffusion model for object detection. In *International Conference on Computer Vision*, 2023b.

Ting Chen, Saurabh Saxena, Lala Li, David Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *International Conference on Learning Representations*, 2022.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023c.

Chengjian Feng, Yujie Zhong, Yu Gao, Matthew Scott, and Weilin Huang. TOOD: Task-aligned one-stage object detection. In *International Conference on Computer Vision*, 2021.

Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. In *International Conference on Computer Vision*, 2021.

Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, and Badong Chen. Salience DETR: Enhancing detection transformer with hierarchical salience filtering refinement. In *Conference on Computer Vision and Pattern Recognition*, 2024.

Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen. DEIM: DETR with improved matching for fast convergence. In *Conference on Computer Vision and Pattern Recognition*, 2025.

Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. DETRs with hybrid matching. In *Conference on Computer Vision and Pattern Recognition*, 2023.

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel Ni, and Lei Zhang. DN-DETR: Accelerate DETR training by introducing query DeNoising. In *Conference on Computer Vision and Pattern Recognition*, 2022.

Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. In *International Conference on Learning Representations*, 2024.

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 *European Conference on Computer Vision*, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, 2017.

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. In *International Conference on Learning Representations*, 2022.

Shilong Liu, Tianhe Ren, Jiayu Chen, Zhaoyang Zeng, Hao Zhang, Feng Li, Hongyang Li, Jun Huang, Hang Su, Jun Zhu, and Lei Zhang. Detection transformer with stable matching. In *International Conference on Computer Vision*, 2023.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *International Conference on Computer Vision*, 2021.

Youssef Ahmed Nawar, Mohamed Badran, and Marwan Torki. DiffuDETR: Rethinking detection transformers with denoising diffusion process. In *International Conference on Learning Representations*, 2026.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *International Conference on Machine Learning*, 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, 2024.

Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-FINE: Redefine regression task of DETRs as fine-grained distribution refinement. In *International Conference on Learning Representations*, 2025.

Yifan Pu, Weicong Liang, Yiduo Hao, Yuhui Yuan, Yukang Yang, Chao Zhang, Han Hu, and Gao Huang. Rank-DETR for high quality object detection. In *Advances in Neural Information Processing Systems*, 2023.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

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 *Conference on Computer Vision and Pattern Recognition*, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse R-CNN: End-to-end object detection with learnable proposals. In *Conference on Computer Vision and Pattern Recognition*, 2021.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *International Conference on Computer Vision*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for transformer-based detector. In *AAAI Conference on Artificial Intelligence*, 2022.

Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient DETR: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021.

Chang-Bin Zhang, Yujie Zhong, and Kai Han. Mr. DETR: Instructive multi-route training for detection transformers. In *Conference on Computer Vision and Pattern Recognition*, 2025.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023.

Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Chuyang Zhao, Yifan Sun, Wenhao Wang, Qiang Chen, Errui Ding, Yi Yang, and Jingdong Wang. MS-DETR: Efficient DETR training with mixed supervision. In *Conference on Computer Vision and Pattern Recognition*, 2024a.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Conference on Computer Vision and Pattern Recognition*, 2024b.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

Zhuofan Zong, Guanglu Song, and Yu Liu. DETRs with collaborative hybrid assignments training. In *International Conference on Computer Vision*, 2023.

A Implementation Details

Architecture We implement PriorDiffuDINO on the basis of DiffuDINO, and Table 5 lists its hyper-parameters. The backbone is a ResNet-50 or ResNet-101 initialized from the ImageNet weights of torchvision, with frozen batch normalization and a frozen stem. The outputs of the last three stages and one additional downsampled level form four feature scales, projected to 256 channels. The encoder and the decoder have six layers each, with 8 attention heads and a feed-forward width of 2048, and there are 900 queries. The content part of the main queries is a learned embedding, and their reference points are built from the top-k encoder proposals as in Section 3.2. The noise level is embedded sinusoidally as $\tau\,(T-1)$ with $T = 100$ and mapped by a two-layer MLP to 1024 channels. Every decoder layer derives the FiLM coefficients of Equation 3 from this embedding and applies them to the query features before its self-attention and before its cross-attention. The contrastive denoising group uses 100 denoising queries with a label noise ratio of 0.5 and a box noise scale of 1.0, and is conditioned on $\tau = 1$.

Training Algorithm 1 lists one training iteration with the two training steps of Section 3.2. The decoder processes the contrastive denoising group and the main group of 900 queries in one sequence, isolated by an attention mask. The proposal head keeps DINO's encoder loss on the clean top-k proposals. Matching uses the costs 2, 5 and 2 for class, L1 and GIoU. The classification loss is the matchability-aware loss of Equation 4 with $\gamma = 1.5$. The loss weights are 2 for classification, 5 for L1 and 2 for GIoU, with a classification weight of 1 for the contrastive denoising group, and

Algorithm 1 Training with self-trajectory denoising (one image).

Require: encoder memory M, the 900 top-scoring proposals b, ground truth G, learnable content queries q, $\sigma_{\max} = 0.5$

1: $\mathcal{L}_{\text{enc}} \leftarrow$ encoder loss of DINO on the clean proposals b
2: $\sigma_1 \sim U(0, \sigma_{\max})$, $\epsilon_1 \sim \mathcal{N}(0, I)$, $r^{(1)} \leftarrow \text{sigmoid}\big(\text{logit}(b) + \sigma_1 \epsilon_1\big)$ ▷ step 1 starts from the noisy prior
3: $(r_{\text{dn}}, c_{\text{dn}}) \leftarrow$ contrastive denoising group, noised copies of G with label embeddings as content
4: $(\hat{b}^{(1)}, \hat{y}^{(1)}, h^{(1)}) \leftarrow D\big([r^{(1)}; r_{\text{dn}}], [q; c_{\text{dn}}]; [\sigma_1; 1], M\big)$ ▷ groups isolated by an attention mask
5: main group: Hungarian matching of $(\hat{b}^{(1)}, \hat{y}^{(1)})$ to G with costs $2/5/2$ (class, L1, GIoU), $\mathcal{L}^{(1)}_{\text{main}} \leftarrow 2\,\mathcal{L}_{\text{MAL}} + 5\,\mathcal{L}_{\text{L1}} + 2\,\mathcal{L}_{\text{GIoU}}$ on every decoder layer
6: denoising group: fixed assignment of positive copies to their ground truth and negative copies to background, $\mathcal{L}^{(1)}_{\text{dn}} \leftarrow 1\,\mathcal{L}_{\text{MAL}} + 5\,\mathcal{L}_{\text{L1}} + 2\,\mathcal{L}_{\text{GIoU}}$
7: $\hat{b}^{(1)} \leftarrow \text{sg}(\hat{b}^{(1)})$, $h^{(1)} \leftarrow \text{sg}(h^{(1)})$ ▷ stop gradient, main group, last decoder layer
8: $\sigma_2 \sim U(0, \sigma_{\max})$, $\epsilon_2 \sim \mathcal{N}(0, I)$, $r^{(2)} \leftarrow \text{sigmoid}\big(\text{logit}(\hat{b}^{(1)}) + \sigma_2 \epsilon_2\big)$ ▷ step 2 re-noises its own prediction
9: $(r'_{\text{dn}}, c'_{\text{dn}}) \leftarrow$ contrastive denoising group rebuilt with fresh noise
10: $(\hat{b}^{(2)}, \hat{y}^{(2)}, h^{(2)}) \leftarrow D\big([r^{(2)}; r'_{\text{dn}}], [h^{(1)}; c'_{\text{dn}}]; [\sigma_2; 1], M\big)$ ▷ same weights, content $= h^{(1)}$
11: $\mathcal{L}^{(2)}_{\text{main}}$, $\mathcal{L}^{(2)}_{\text{dn}} \leftarrow$ as in lines 5–6, with the matching recomputed for step 2
12: $\mathcal{L} \leftarrow \mathcal{L}_{\text{enc}} + \mathcal{L}^{(1)}_{\text{main}} + \mathcal{L}^{(1)}_{\text{dn}} + \mathcal{L}^{(2)}_{\text{main}} + \mathcal{L}^{(2)}_{\text{dn}}$ and update the parameters

they apply to all six decoder layers. The second training step takes the boxes and the query states of the last decoder layer of the first step with the gradient stopped, draws $\sigma_2 \sim U(0, 0.5)$ per image and rebuilds the contrastive denoising group. It shares the decoder weights and the losses of the first step with weight 1.

Data The plain augmentation is that of DINO, namely random horizontal flip and multi-scale resizing of the shorter side to 480–800 pixels with the longer side at most 1333, or a branch with a random crop. Dense augmentation applies, independently with probability 0.5 each, a 2×2 mosaic of four images resized to a shorter side of 320–448 pixels and a mixup with a second image at a blend ratio drawn from $U(0.3, 0.7)$. A sample that draws neither goes through the plain augmentation. The dense loader is active between 5% and 50% of the iterations (8,871 to 88,715 on COCO) and the plain loader elsewhere. At test time the shorter side is resized to 800 pixels.

Optimization We use AdamW with a learning rate of 10^{-4}, 10^{-5} for the backbone, $\beta = (0.9, 0.999)$, a weight decay of 10^{-4} and gradient clipping at norm 0.1. The batch contains 8 images, and 12 epochs amount to 177,431 iterations on COCO, without warm-up. The learning rate is divided by 10 after 11/12 of the iterations. An exponential moving average of the parameters and the buffers with decay 0.999 is updated at every iteration and used only for evaluation, as in the released configuration of DiffuDINO. The ResNet-50 COCO model is evaluated with these averaged weights.

Inference All 900 queries start from the noised proposals with $\sigma_1 = \sigma_{\max} = 0.5$ and follow the schedule of Section 3.2. Every later step adds fresh noise around the previous prediction, carries the previous query states as content and uses $\tau = \sigma_i$. The last step is the output, from which the 300 highest-scoring (query, class) pairs are kept without NMS. The entries of Table 1 use $k = 3$. Algorithm 2 lists the steps. The inference noise is seeded by the image id, and every number is reproducible.

LVIS The LVIS models use the same protocol with 1203 classes. Repeat-factor sampling with threshold 10^{-3} (Gupta et al., 2019) is applied to both the plain and the dense loader. Twelve epochs of the 100,170 training images amount to 150,255 iterations, and the evaluation on the 19,809 validation images keeps 300 detections per image. The ResNet-101 model on LVIS is trained with a micro-batch of 4 and two accumulation steps, which keeps the effective batch at 8.

Algorithm 2 Inference with k denoising steps.

Require: image, number of steps k, $\sigma_{\max} = 0.5$, learnable content queries q

$M \leftarrow \text{encoder}(\text{image})$, $\hat{b}^{(0)} \leftarrow$ the 900 top-scoring proposals, $h^{(0)} \leftarrow q$

for $i = 1, \dots, k$ **do**

  $\sigma_i \leftarrow \sigma_{\max}\,(1 - \frac{i-1}{k})$, $\epsilon_i \sim \mathcal{N}(0, I)$ ▷ noise seeded by the image id

  $r^{(i)} \leftarrow \text{sigmoid}\big(\text{logit}(\hat{b}^{(i-1)}) + \sigma_i \epsilon_i\big)$ ▷ Equation 1

  $(\hat{b}^{(i)}, \hat{y}^{(i)}, h^{(i)}) \leftarrow D\big(r^{(i)},\, h^{(i-1)};\, \sigma_i,\, M\big)$ ▷ Equation 2, main group only

end for

Ensure: $(\hat{b}^{(k)}, \hat{y}^{(k)})$ of the last step, the 300 highest-scoring (query, class) pairs, with no box renewal, no merging of steps and no NMS

Table 5: Hyper-parameters of PriorDiffuDINO. The COCO values are for the ResNet-50 model, and the differences for LVIS are given in the text.

Backbone / feature scales / queries	ResNet-50 (ImageNet), 4 scales, 900
Encoder / decoder layers, width, heads, FFN	6 / 6, 256, 8, 2048
Contrastive denoising	100 queries, label noise 0.5, box noise 1.0, $\tau = 1$
Matching cost (class / L1 / GIoU)	2 / 5 / 2
Loss weights (class / L1 / GIoU / CDN class)	2 / 5 / 2 / 1
Matchability-aware loss	$\gamma = 1.5$
Noisy prior	$\sigma \sim U(0, 0.5)$ in logit space, $\tau = \sigma$
Second training step	$\sigma_2 \sim U(0, 0.5)$, stop-gradient, weight 1
Time embedding	sinusoidal, $T = 100$, MLP to 1024, FiLM per layer
Dense augmentation	mosaic 0.5, mixup 0.5, iterations 5%–50%
Optimizer	AdamW, lr 10^{-4} (backbone $\times 0.1$), wd 10^{-4}
Batch / iterations / LR drop	8 / 177,431 (12 epochs) / at 11/12
Gradient clipping / warm-up	0.1 / none
EMA	decay 0.999, evaluation only
Inference	k steps, $\sigma_i = 0.5\,(1 - \frac{i-1}{k})$, top-300, no NMS

B Detailed Comparison with DINO, DiffuDETR and DiffuDINO

Table 6 lists, component by component, how PriorDiffuDINO differs from DINO (Zhang et al., 2023) and from DiffuDETR and DiffuDINO (Nawar et al., 2026). The entries for DiffuDETR and DiffuDINO follow their paper and, where the paper is silent, the released code of DiffuDINO. The sampler of DiffuDETR is not released, so its box renewal entry is marked as not described. In the released code, boxes whose highest class score falls below a threshold equal to the next timestep divided by 100 are replaced by fresh Gaussian samples between steps. The predictions of all steps are pooled and filtered by NMS at an IoU threshold of 0.8 whenever more than one step is run.

C Additional Results

Table 7 reports all COCO metrics of the ResNet-50 model for one to ten denoising steps under the default inference of Section 3.2, evaluated with the EMA weights. AP lies between 51.2 and 51.4 for every k, AP_S between 34.8 and 35.2, and AP_L between 66.4 and 66.9.

Table 8 extends the w/o Prior ablation of Table 3 to more steps. Starting from the encoder proposals, the accuracy is nearly constant. Starting from Gaussian boxes, the first step reaches 43.5 AP, and the later steps raise it to 50.5, 51.2 and 51.4 AP with three, five and ten steps. With ten steps the two starting points differ by 0.1 AP.

Table 6: Component comparison. Each column is one detector, and each row is one design choice.

	DINO	DiffuDETR	DiffuDINO	PriorDiffuDINO
Starting point in training	top-k encoder proposals	ground-truth boxes noised by the forward process, t uniform in $\{0, \dots, 99\}$	as DiffuDETR	top-k encoder proposals noised with $\sigma \sim U(0, 0.5)$
Starting point at inference	top-k encoder proposals	boxes drawn from $\mathcal{N}(0, I)$	as DiffuDETR	the same noised proposals, $\sigma_1 = 0.5$
Timestep condition	none	embedded timestep t	as DiffuDETR	noise level $\tau = \sigma$
Trained on own predictions	no	no	no	yes, second training step on its own trajectory
Box renewal / NMS at inference	none	not described	renewal below a score threshold, pooling over steps and NMS at 0.8	none
Classification loss	focal	focal	focal	matchability-aware
Augmentation	flip, multi-scale, crop	default of detrex	default of detrex	DINO's, plus mosaic and mixup for 45% of training
Epochs to reported result (ResNet-50, COCO)	12 / 36	50	50	12

Table 7: PriorDiffuDINO with ResNet-50 on the COCO 2017 validation set for $k = 1, \dots, 10$ denoising steps, starting from the noised encoder proposals with $\sigma_1 = 0.5$ and the schedule $\sigma_i = \sigma_{\max}(1 - \frac{i-1}{k})$.

Steps k	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	51.2	68.1	56.0	35.0	54.9	66.4
2	51.3	68.3	56.2	35.2	54.9	66.8
3	51.3	68.4	56.1	35.2	55.0	66.9
4	51.2	68.3	56.0	35.0	54.8	66.9
5	51.4	68.5	56.2	35.1	55.0	66.9
6	51.3	68.6	56.2	35.1	55.0	66.7
7	51.4	68.5	56.1	35.1	54.9	66.8
8	51.4	68.7	56.3	34.8	54.9	66.8
9	51.4	68.6	56.2	35.2	54.9	66.6
10	51.3	68.5	56.1	35.1	54.9	66.5

Table 8: Starting point and number of denoising steps. PriorDiffuDINO with ResNet-50 on the COCO 2017 validation set, starting either from the noised encoder proposals or from Gaussian boxes at inference, with the same weights and the same schedule.

Starting point	Steps k	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Encoder proposals	1	51.2	68.1	56.0	35.0	54.9	66.4
	3	51.3	68.4	56.1	35.2	55.0	66.9
	5	51.4	68.5	56.2	35.1	55.0	66.9
	10	51.3	68.5	56.1	35.1	54.9	66.5
Gaussian boxes	1	43.5	57.1	47.3	23.1	49.3	66.1
	3	50.5	67.1	55.2	33.5	54.7	66.7
	5	51.2	68.3	55.9	34.9	54.9	67.0
	10	51.4	68.6	56.1	35.0	54.9	66.7