Diffuse&Refine: Intrinsic Knowledge Generation and Aggregation for Incremental Object Detection

Jianzhou Wang[1,2], **Yirui Wu**[1,2*], **Lixin Yuan**[1,2], **Wenxiao Zhang**[1,2], **Jun Liu**[3], **Junyang Chen**[4], **Huan Wang**[5], **Wenhai Wang**[6]
[1]College of Computer Science and Software Engineering, Hohai University
[2]Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University
[3]School of Computing and Communication, Lancaster University
[4]College of Computer Science and Software Engineering, Shenzhen University
[5]College of Informatics, Huazhong Agricultural University
[6]Multimedia Laboratory, The Chinese University of Hong Kong
{wangjianzhou, wuyirui, yuanlixin}@hhu.edu.cn, wenxxiao.zhang@gmail.com, j.liu81@lancaster.ac.uk, junyangchen@szu.edu.cn, hwang@mail.hzau.edu.cn, whwang@ie.cuhk.edu.hk

Abstract

Incremental Object Detection(IOD) targets at progressively extending capability of object detectors to recognize new classes. However, representation confusion between old and new classes leads to catastrophic forgetting. To alleviate this problem, we propose DiffKA, with intrinsic knowledge generated and aggregated by forward and backward diffusion, gradually establishing rigid class boundary. With incremental streaming data, forward diffusion spreads information to generate potential inter-class associations among new- and old-class prototypes within a hierarchical tree, named as Intrinsic Correlation Tree(ICT), to store intrinsic knowledge. Afterwards, backward diffusion refines and aggregates the generated knowledge in ICT, explicitly establishing rigid class boundary to mitigate representation confusion. To keep semantic consistency with extreme IOD settings, we reorganize semantic relevance of old- and new-class prototypes in paradigms to adaptively and effectively update DiffKA. Experiments on MS COCO dataset show DiffKA achieves state-of-the-art performance on IOD tasks with significant advantages.

1 Introduction

Humans expect learning models to continually handle new tasks without forgetting old ones. However, models suffer from catastrophic forgetting in incremental learning, facing representation confusion between old and new classes (see Fig. 1(a)). Due to the independent and identical input of new classes [Gurbuz *et al.*, 2024], feature representations of old classes would vary to mis-match the correct ones, causing semantic drift to exacerbate inter-class confusion (see Fig. 1(b)). These problems have sparked Incremental Object Detection(IOD), requiring to acquire knowledge on extracting representations of new classes and maintaining old ones.

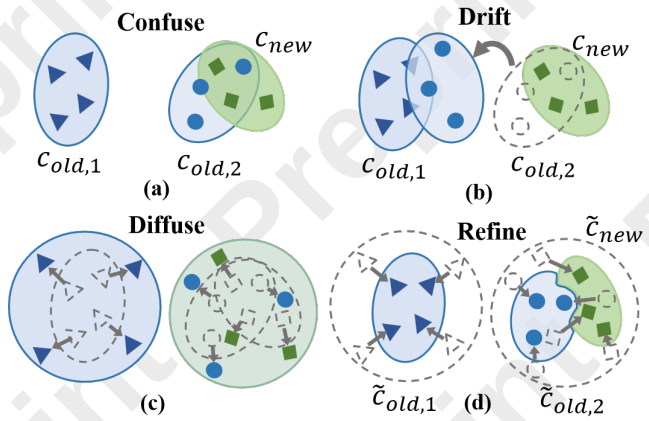

Figure 1: (a) In IOD tasks, new class c_{new} often confuse with old class c_{old} in semantic representation, causing catastrophic forgetting. (b) Old-class feature drifts to exacerbate inter-class confusion between $c_{old,1}$ and $c_{old,2}$. (c) DiffKA simulates forward diffusion to generate inter-class associations, i.e., intrinsic knowledge. (d) DiffKA simulates backward diffusion to refine and aggregate knowledge, thus establishing rigid class boundary between $\tilde{c}_{new}$ and $\tilde{c}_{old,2}$ to mitigate catastrophic forgetting.

Existing methods address problems with either Knowledge Distillation(KD) or Replay [Li and Hoiem, 2016]. KD stores old-class knowledge by incorporating parametric neurons with low-level semantics, working as a implicit knowledge transfer and requiring further abstraction [Kang *et al.*, 2023]. Meanwhile, Replay covers massive old data with few key exemplars, only memorizing salient part with losing old-class knowledge details. Despite achieving notable advances, they struggle to simultaneously keep old-class knowledge and update new-class one regarding incremental data stream. We argue their respective drawbacks can be alleviated by introducing knowledge to generate rigid boundary between old- and new-class representations. Such knowledge not only re-

*Corresponding author.

solves inter-class confusion with explicit representation partition among different classes, but also mitigates semantic drift by maintaining details of old-class knowledge with anchor-like representations corresponding to rigid class boundary.

With the emergence of advanced multi-modal large language models(MLLMs), there has been a further exploration by importing external knowledge [Junsu *et al.*, 2024]. However, by gradually injecting knowledge accompanying with incremental stream, the retrieving knowledge from external databases may not have strong semantic relevance with either old or new classes, progressively misleading into areas that MLLMs originally excelled. Therefore, involving intrinsic knowledge learned from incremental data itself should be emphasized, establishing rigid class boundary to boost performance with respect to catastrophic forgetting.

We thus propose DiffKA, an intrinsic knowledge generation and aggregation method, which gradually establishes rigid class boundary via diffusion to alleviate catastrophic forgetting. Inspired by high-quality reconstruction and strong associative ability [Rombach *et al.*, 2022], we develop a diffusion-based view for generation and aggregation of intrinsic knowledge, representing them as forward and backward process respectively. Therefore, we first simulate forward diffusion to spread information gained from streaming data, generating potential inter-class associations as wide and deep as possible(see Fig. 1(c)). Specifically, forward diffusion integrates representations of old- and new-class prototypes regarding different learning phases and semantic levels, building hierarchical correlations with an organized tree named as Intrinsic Correlation Tree(ICT). Based on ICT storing intrinsic knowledge, backward diffusion is simulated by refining and aggregating knowledge, explicitly establishing rigid class boundaries in semantic space(see Fig. 1(d)). Specifically, extra supervised information generated by forward diffusion, such as labels, correlations and so on, assists class representations to progressively and optimally converge in semantic space as detailed knowledge. By aggregating fine-grained knowledge in different learning phases and different semantic levels, DiffKA stabilizes convinced correlations among both old and new classes for IOD task.

We further enhance the updating of ICT considering dynamic environment of IOD. ICT still requires emphasis on that intrinsic knowledge should focus on cross-phase and cross-level semantic consistency, in case that imbalanced, few or extreme inputs cause disorders of semantic space. So we leverage rearranging semantic relevance in paradigms to keep semantic consistency. Regarding cross-phase inconsistency, we adaptively enhance representation capability of new-class prototypes to adjust forward diffusion, thus decreasing uncertainty of defining class boundary with few or imbalanced input. Regarding cross-level inconsistency, we employ different types of tree structure adjustment in backward diffusion based on variant scenarios, thus avoiding potential conflicts in intrinsic knowledge. With updating promotion, DiffKA ensures rigid class boundary with excellent reliability and generality, even under extreme settings of IOD. Overall, the main contributions are three-fold:

- To avoid catastrophic forgetting in IOD, we propose DiffKA, with intrinsic knowledge generation and aggregation performed by forward and backward diffusion, building Intrinsic Correlation Tree(ICT) to establish rigid boundary between old and new classes.
- Regarding cross-phase and cross-level inconsistency within ICT, we enhance its adaptively and effectively updating via rearranging semantic relevance in paradigms.
- Experimental results demonstrate DiffKA achieves state-of-the-art performance with significant advantages.

2 Related Work

2.1 Incremental Object Detection

To retain knowledge, IOD methods utilize either knowledge distillation [Feng *et al.*, 2022; Kang *et al.*, 2023; Liu *et al.*, 2023a] or replay [Liu *et al.*, 2023a; Liu *et al.*, 2023b; Junsu *et al.*, 2024]. ERD [Feng *et al.*, 2022] performs elastic distillation on the response of classification and regression heads to address class imbalance. Regarding destruction of semantic space as cause of catastrophic forgetting, [Kang *et al.*, 2023] dynamically distills between-class and within-class semantics to prevent forgetting.

For replay-based methods, CL-DETR [Liu *et al.*, 2023a] selectively stores exemplars of old data to memorize previous learned samples. ABR [Liu *et al.*, 2023b] embeds old instances into new backgrounds to reduce shift. SDDGR [Junsu *et al.*, 2024] generates virtual images via controllable generators, eliminating original-data reliance.

2.2 Diffusion Model

Diffusion models [Ho *et al.*, 2020; Nichol and Dhariwal, 2021; Rombach *et al.*, 2022] are famous for impressive generative capability. LDM [Rombach *et al.*, 2022] diffuses in latent space for high-quality outputs. DiffusionDet [Chen *et al.*, 2023] reformulates box prediction as a denoising process, while DFDD [Wu *et al.*, 2023] uses forward diffusion of OOD features and reverse recovery to sharpen feature discrimination. DiffKA leverages diffusion to generate and aggregate intrinsic knowledge on rigid class boundary, expanding diffusion to aid abstraction of knowledge.

3 Methodology

3.1 Task Description

We follow the stricter IOD protocol of [Liu *et al.*, 2023a]. Formally, given a categories set $\mathcal{C} = \{1, 2, \ldots, C\}$, we represent a dataset as $\mathcal{D} = \{(x, y)\}$, where x denotes an image sample, and $y = \{(b, c)\}$ is the annotation set to indicate the bounding boxes b and the set of category labels $c \in \mathcal{C}$ corresponding to objects within the image. $\mathcal{D}$ and $\mathcal{C}$ are then divided into M disjoint subsets based on the number of training phases, where $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_M$ and $\mathcal{C} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_M$. For phase m, we focus on samples in $\mathcal{D}_m$, where y only contains annotations for objects of Class $\mathcal{C}_m$ and discard the others. In the mth training phase, the model need recognize the objects in $\mathcal{C}_m$ with $\mathcal{D}_m$, while retaining the capability to recognize all previously learned categories $\mathcal{C}_{1:m-1}$. Notably, the model can observe all samples of categories in former phases, while samples of specified categories are annotated in the ith phase.

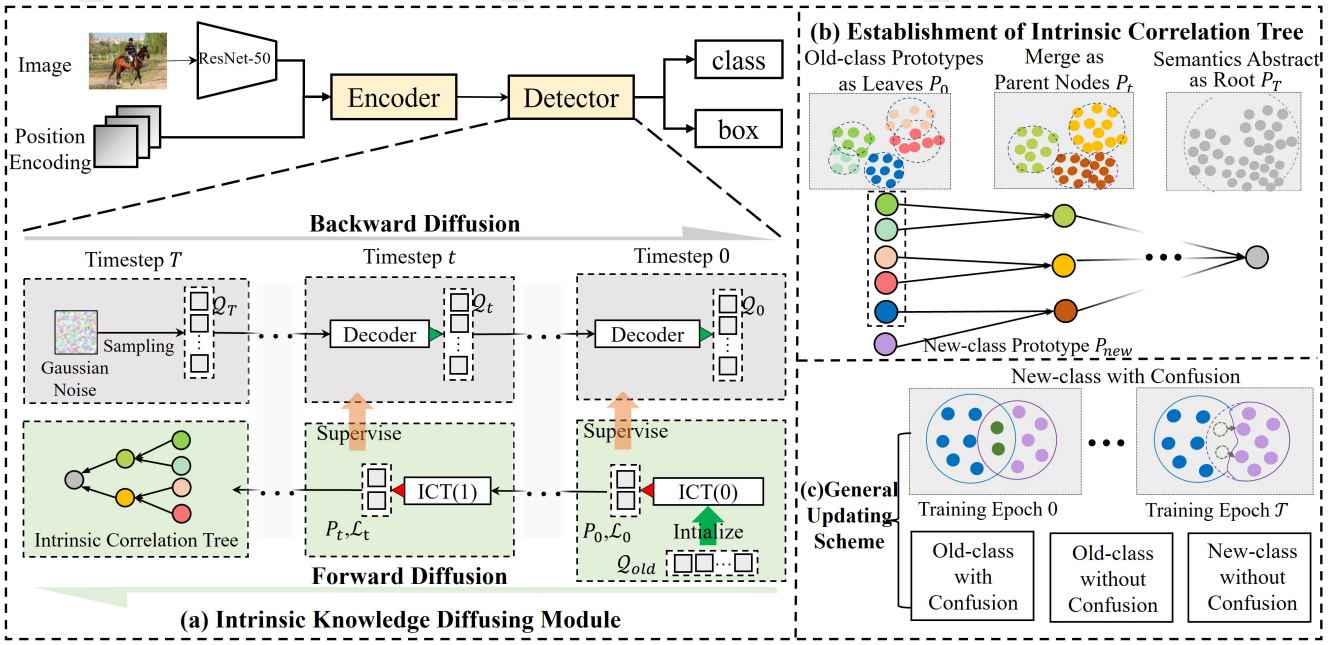

Figure 2: Structure overview of DiffKA, which consists of (a) Intrinsic Knowledge Diffusing Module, (b) Establilshment of Intrinsic Correlation Tree and (c) General Updating Scheme. In forward diffusion, prototypes P_t are extracted to establish ICT with hierarchical semantic levels. In backward diffusion, the detector gradually refines the coarse object queries Q_t supervised by ICT.

3.2 Overview

As shown in Fig. 2, we present the structure overview of DiffKA. Inspired by the end-to-end detector, i.e., Deformable-DETR [Zhu *et al.*, 2021], we first feed images into ResNet-50 backbone network to extract feature maps, which are then fed into encoder for position encoding. After encoding, the detector would predict a set of object queries. Finally, class and bounding box head would predict classes and positions of all objects based on queries. It's noted that detector is pre-trained with old classes, while we perform incremental learning to detect new classes with the proposed intrinsic knowledge diffusing module and enhanced updating scheme.

In intrinsic knowledge diffusing module(see Fig.2(a)), we simulate diffusion to gradually build Intrinsic Correlation Tree(ICT), with forward and backward diffusion generating and aggregating intrinsic knowledge, i.e., inter-class associations in semantic space. During forward diffusion, at Forward Timestep 0, we adopt detector to predict old-class instances $\mathcal{S}_{old} = \{S_i | i = 1, ..., K\}$ with K classes of pre-learned semantic information. Based on $\mathcal{S}_{old}$, we further extract set of informative K prototypes $P_0 = \{P_{i,0} \mid i = 1, ..., K\}$, which are fed to build ICT as initialization of its leave nodes. As shown in Fig. 2(b) to illustrate the establishment of ICT, P_0 gradually merged to form its N parent nodes $P_t = \{P_{i,t} \mid i = 1, ..., N; t = 1, ..., T-1\}$ by step-wisely adding noise for simulation of forward diffusion, which explores potential inter-class associations as wide and deep as possible. Meanwhile, we progressively optimize towards non-overlap and geometrically distinct properties of P_t with reasonable prototype merging rules. During merging, we also generate label set $\mathcal{L}_t = \{\mathcal{L}_{i,t} \mid i = 1, ..., N; t = 1, ..., T-1\}$ for each parent node by containing labels of all leave nodes. It's noted that we use the joint representation of labels other than individual category label, which exhibits the enhanced semantics of multi-label prototypes stored in ICT. Finally, we could abstract all old-class semantic representations as root of ICT P_T, aggregating intrinsic knowledge in a progressive optimizing manner.

During backward diffusion, we refine and aggregate knowledge provided by ICT. At Backward Timestep T, we first sample sets of object queries $\mathcal{Q}_T$ from Gaussian noise. Based on $\mathcal{Q}_T$, we adopt decoders to step-wisely refine $\mathcal{Q}_T$ with the intrinsic semantic knowledge generated in forward diffusion, i.e., additional supervised information referring to $\mathcal{L}_{0:t-1}$ and $P_{0:t-1}$ stored in ICT, progressively generating denoised and semantic enriched object queries $\mathcal{Q}_t$.

Considering incremental streaming, we propose a general updating scheme to deal with four scenarios (see Fig. 2(c)). Facing old-class samples with representation confusion, we would abandon them due to lack of ground-truth labels. Facing old-class samples without confusion, we would update their corresponding old-class prototypes P_0 with the inputting samples. Facing new-class samples without confusion, we would build a new prototype P_{new} to update ICT as a leaf node for further forward diffusion. Facing new-class samples with confusion to cause catastrophic forgetting as shown in Fig. 1(a), we gradually move semantic representations of new-class samples via backward diffusion, generating rigid class boundary to avoid confusion.

In enhanced updating scheme, we solve cross-phase and cross-level semantic inconsistency with different strategies. Due to rich intrinsic knowledge of old-class learned and

stored in previous phases, the detector would generate imbalanced number of new-class and old-class instances to cause semantic disorders. By calculating weight based on new-class object queries $\mathcal{Q}_{new}$ and old-class ones $\mathcal{Q}_{old}$, we adaptively adjust forward diffusion to generate balanced prototype representation P_t. Meanwhile, cross-level semantic inconsistency leads to conflicting confusions between new-class object query $\mathcal{Q}_{new}$ and old-class prototypes P_{t-1}, P_t generated by decoders at Timesteps $t-1$, t. Such confusions result in unconvinced and terribly organized intrinsic knowledge stored in ICT. By determining whether P_t and P_{t-1} exists parent-son relationship, we would either update ICT with general updating scheme if existing, or build new-class prototypes P_{new} from the conflicting level to higher levels as well as from Timestep $t+1$ to T if not.

3.3 Intrinsic Knowledge Diffusion Module

In this subsection, we focus on generating the ICT within Intrinsic Knowledge Diffusion module. Specifically, we employ forward diffusion to extract intrinsic knowledge from the incremental data stream to construct ICT. Within the ICT, the detector utilizes backward diffusion to refine information and establish rigid boundaries among all classes.

Forward Diffusion. At Forward Timestep 0, we feed detector Φ with new samples $\mathcal{D}_m$, to predict all instances of the learned old classes $\mathcal{S}_{old} = \Phi(\mathcal{D}_m)$. Based on $\mathcal{S}_{old}$, we extract set of informative K prototypes $P_0 = \{P_{i,0} \mid i = 1, ..., K\}$, where we define $P_{i,0}$ as the mean distribution center of all instances belonging to class i, i.e., $P_{i,0} = \frac{1}{|\mathcal{S}_i|}\sum \mathcal{S}_i$. As demonstrate by [Kothapalli, 2023], all instances of the same class tend to form cluster-like distribution, leading $P_{i,0}$ to be efficient in representing semantic information. Afterwards, P_0 are used to build ICT with each prototype initialized as its leaf node.

In Forward Diffusion period $t \in \{0, \ldots, T-1\}$, step-wise Gaussian noise $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ is added to P_t, progressively blurring semantic information of prototypes over time:

$$P_{t+1} = \sqrt{\bar{\alpha}_t} P_t + \sqrt{(1-\bar{\alpha}_t)}\mathcal{N}(0, \sigma_t^2), \tag{1}$$

where $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i$ controls strength of the added noise. $\alpha_i = 1 - \beta_i$, where β_i represents the variable referring to noise variance gradient illustrated in [Ho *et al.*, 2020]. With noise increasing, class prototypes overlap at first, and then the most similar prototypes start to merge. To achieve their non-overlap and geometrically distinct properties, we propose to optimize P_t with merging rules, which is defined to search two prototypes $P_{i,t}$ and $P_{j,t}$ for merging with the minimum Kullback-Leibler Divergence:

$$(i, j) = argmin_{i,j} \sum_{i,j=1}^{K} P_{i,t} \ln \frac{P_{i,t}}{P_{j,t}}. \tag{2}$$

To prevent over or insufficient merging, we further define the number of parent nodes $N_t = \lceil |\mathcal{C}| \times \eta^t \rceil$, where $\lceil \cdot \rceil$ donates the rounding up function, and $\eta \in [0, 1]$ is a hyperparameter to control the rate of prototype merging. It's noted that we adopt a higher η, since it result in fewer fusions at each timestep, thus forcing DiffKA to focus on the easily confused classes in forward diffusion.

With P_0 iteratively merging to less prototypes, we achieve an organized and hierarchical tree named as ICT, which captures intrinsic knowledge with semantic correlation established during forward diffusion. Similar with other diffusion models, the forward diffusion involves none of learnable parameters.

Backward Diffusion. We refine knowledge provided by ICT iteratively in backward diffusion. At Backward Timestep T, a fixed number of object queries $\mathcal{Q}_T$ are sampled from Gaussian noise. Based on $\mathcal{Q}_T$, we adopt decoder ϕ_t to stepwisely refine $\mathcal{Q}_t$ at Backward Timestep t with the corresponding intrinsic semantic knowledge stored in ICT, i.e., $\mathcal{L}_t$ and P_t, and previous $\mathcal{Q}_{t+1}$:

$$\mathcal{Q}_t = \phi_t(\mathcal{Q}_{t+1}, F, \mathcal{L}_t, P_t), \tag{3}$$

where F is the feature encoding of the inputting image. Based on $\mathcal{Q}_t$, we calculate the unified loss function L_{CID} for refinement:

$$L_{CID} = \sum_{t=0}^{T-1} \sum_{i=1}^{K} \mathbf{1}_{i \in \mathcal{L}_t} \cdot (L_{cls} + L_{box} + L_{pc}), \tag{4}$$

where $\mathbf{1}_{i \in \mathcal{L}_t}$ donates that we only compute L_{CID} for object queries if its predicted class exists in set $\mathcal{L}_t$. Unlike [Liu *et al.*, 2023a] which treats background class as negative instances for supervision, we filter instances with labels not existing in $\mathcal{L}_t$, thus emphasizing to retain the missing annotations as well as additional information in incremental environment. Furthermore, we achieve L_{CID} with tree-structured labels $\mathcal{L}_t$ rather than individual class labels, ensuring the detector to focus on the annotated classes and simultaneously avoiding to misclassify unannotated objects as background.

Specifically, classification loss L_{cls}, bounding box prediction lossL_{box} and prototype constraint loss L_{pc} are defined as:

$$\begin{cases} L_{cls} = \sum_{c \in \mathcal{L}} -p_i(c) \log \hat{p}_{\hat{\sigma}_i}(c) \\ L_{box} = \gamma_1 f_{IoU}(\hat{b}_{\hat{\sigma}_i}, b_i) + \gamma_2 \|\hat{b}_{\hat{\sigma}_i} - b_i\|_1 \\ L_{pc} = \|\mathcal{Q}_t, P_t\| = \mathcal{Q}_t \ln \frac{\mathcal{Q}_t}{P_t}, \end{cases} \tag{5}$$

where $p_i(c)$ denotes one-hot label encoding for class c, $\hat{p}_{\hat{\sigma}_i}(c)$ represents the confidence of prediction, γ_1 and γ_2 are hyperparameters, function $f_{IoU}()$ calculates intersection of union for the generated bounding boxes, $\hat{b}_{\hat{\sigma}_i}$ and b_i represent predicted and annotated bound boxes respectively, $\|\cdot, \cdot\|_1$ refers to L1 Normalization to measure Euclidean distance between vectors, and $\|\cdot, \cdot\|$ computes the Kullback-Leibler Divergence of two distributions. Note that we obtain $\hat{b}_{\hat{\sigma}_i}$ via binary matching [Carion *et al.*, 2020] to search for the best annotations for object queries.

Specifically, L_{cls} encourages the detector to make high confidence predictions with multiple class labels for a single object, thus representing inter-class semantics within one object. By designing classification loss with multiple labels in backward diffusion, we facilitate the aggregation of shared intrinsic knowledge corresponding to single object instance. L_{box} evaluates the accuracy of bounding box prediction. L_{pc} refers to the additional supervised information extracted from prototypes $P_{0:t-1}$, thus keeping refined queries $\mathcal{Q}_t$ to be consistent with the stored prototypes in ICT.

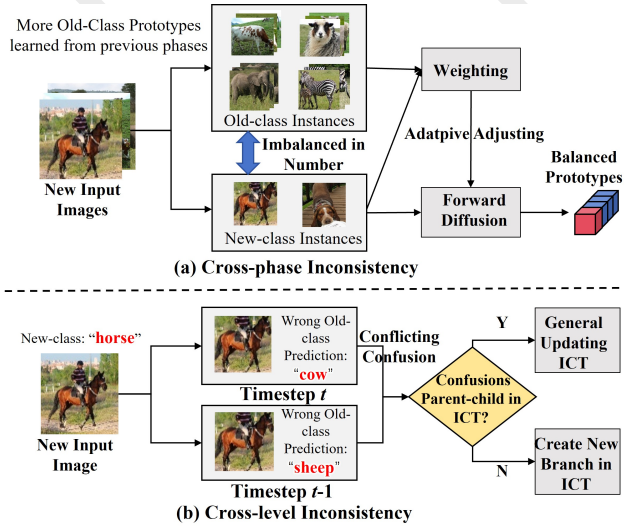

Figure 3: Workflow of the proposed enhanced updating scheme. (a) To solve cross-phase inconsistency caused by imbalance instance numbers between old and new classes, we re-weight their contribution in forward diffusion to generate balanced prototypes. (b) To solve cross-level inconsistency caused by the conflicting confusion, i.e., one new class confuses with two different old classes at different levels in ICT, we either perform general updating or create new branch by analyzing their relationship.

3.4 Enhanced Updating Scheme

In this subsection, we first introduce general updating scheme of ICT with incremental streaming data. Afterwards, we propose enhanced updating scheme to adaptively and efficiently update ICT, facing cross-phase and cross-level inconsistencies induced by extreme settings of IOD.

General Updating Scheme. As shown in Fig. 2(c), the general updating scheme is designed to deal with four scenarios. Focusing on representation confusion between new and old classes to cause catastrophic forgetting, at Backward Timestep 0 of the 1st training epoch for diffusion, we create a new prototype without the confusing samples to update ICT as a leaf node, maintaining non-overlap and geometrically distinct properties of all nodes.

At Backward Timestep 0 of the 2nd training epoch, we use the updated ICT to supervise the generation of new-class prototype with the confusing samples. Since its theory of Markov Chain ensures to effectively perform neighbor searching [Ho *et al.*, 2020], diffusion owns capability to generate the cross samples on class boundary. Therefore, its generated supervision would gradually pull confusing samples in semantic space, until their detection results fits ground-truth labels in training epoch $\mathcal{T}$, thus generating new-class prototype P_{new} with rigid class boundary among old and new classes. In other words, the intrinsic knowledge embedded in ICT guides the way of boundary shifting, gradually moving confusing samples to locate near the boundary with supervised information provided via diffusion.

Enhanced Updating Scheme. Since IOD tasks often involve unstable incremental data streams, DiffKA may encounter extreme inputs with imbalanced class samples across different learning phases, thus causing cross-phase inconsistency in generating intrinsic knowledge during forward diffusion. Meanwhile, semantic information of new-class prototypes may vary greatly at different backward timesteps, leading to cross-level inconsistency. Both inconsistencies disorder the semantic space to exacerbate catastrophic forgetting.

Due to rich intrinsic knowledge of old-class learned and stored in previous phases, the detector would generate imbalanced number of new-class and old-class instances to cause cross-phase inconsistency (see Fig. 3(a)). Since prototype is defined as mean center of instances, few new-class instances rarely have impacts on prototypes stored in ICT P_t, which leads DiffKA to struggle in learning new-class semantic information. We thus rearrange semantic correlations between old and new classes, which weights new-class prototypes to not only boost the efficiency of its knowledge aggregation, but also strengths semantic associations between new-class and old-class prototypes. Given the number of old- and new-class instances, i.e., $|\mathcal{S}_{old}|$ and $|\mathcal{S}_{new}|$, the total number of old and new class, i.e., $|\mathcal{C}_{old}|$ and $|\mathcal{C}_{old}|$, we thus weight new-class instances with the old ones to adjust their impacts on forward diffusion:

$$P_t = \frac{|\mathcal{S}_{old}|}{|\mathcal{C}_{old}|} P_{i,t-1} \oplus \frac{|\mathcal{S}_{new}|}{|\mathcal{C}_{new}|} P_{j,t-1}, i \in \mathcal{C}_{old}, j \in \mathcal{C}_{new}, \quad (6)$$

where $\oplus$ represents the operation of element-wise sum.

Due to the small number of new-class samples, DiffKA obtains unstable representation in semantical space during backward diffusion, which might cause confusions with different levels of old-class prototypes stored in hierarchical ICT (see Fig. 3(b)). To resolve cross-level inconsistency, we define confusion predictions with two criteria: 1) not background class, where the predicted bounding box should coincide with the ground-truth, 2) the confusion occurs between new-class and old-class predictions with high confidence. At Backward Timestep t, We thus achieve the confused predictions U_t form object queries $\mathcal{Q}_t$:

$$U_t = \{(\hat{b}, \hat{c}) \in f_h(\mathcal{Q}_t) \mid f_{GIoU}(\hat{b}, b) \geq \lambda_1, p(\hat{c}) \geq \lambda_2, \hat{c} \in \mathcal{C}_{old}, c \in \mathcal{C}_{new}\}, \quad (7)$$

where $f_h(\cdot)$ is the classification and regression head mapping $\mathcal{Q}_t$ to predicted class label $\hat{c}$ and bounding box $\hat{b}$, function $f_{GIoU}()$ measures the overlap or distance between bounding boxes [Rezatofighi *et al.*, 2019], $p(\hat{c})$ is the prediction confidence of object query for class c, and λ_1 and λ_2 are hyperparameters.

Afterwards, we detect the conflicting confusion by determining whether P_t and P_{t-1} exists parent-son relationship. If exists referring to no conflicting confusion, DiffKA would perform the general updating scheme. If not exists referring to occurring of conflicting confusion, we adaptively adjust ICT to prevent the conflict. More precisely, if a conflict is detected at timestep t, DiffKA would create new prototypes for P_t, which are first connected as children to P_{t-1}, and then created as nodes at higher semantic levels as well as later backward timesteps $t \in \{t+1, t+2, \ldots, T\}$. Therefore, such adjustment would create a new branch in ICT for cross-level inconsistent classes.

Setting	Method	Baseline	Buffer Rate	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
70+10	LwF	Deformable DETR	10%	24.5	36.6	26.7	12.4	28.2	35.2
	RILOD	GFLv1	0%	29.9	45.0	32.0	15.8	33.0	40.5
	SID	GFLv1	0%	34.0	51.4	36.3	18.4	38.4	44.9
	iCaRL	Deformable DETR	10%	35.9	52.5	39.2	19.1	39.4	48.6
	ERD	Deformable DETR	10%	36.9	55.7	40.1	21.4	39.6	48.7
	CL-DETR	Deformable DETR	10%	40.1	57.8	43.7	**23.2**	43.2	52.1
	DiffKA	Deformable DETR	0%	40.3	57.6	**44.1**	22.9	**44.0**	53.3
	DiffKA$^+$	Deformable DETR	10%	**40.5**	**58.4**	**44.1**	22.3	**44.0**	**53.9**
40+40	LwF	Deformable DETR	10%	23.9	41.5	25.0	12.0	26.4	33.0
	RILOD	GFLv1	0%	24.5	37.9	25.7	14.2	27.4	33.5
	SID	GFLv1	0%	32.8	49.0	35.0	17.1	36.9	44.5
	iCaRL	Deformable DETR	10%	33.4	52.0	36.0	18.0	36.4	45.5
	ERD	Deformable DETR	10%	36.0	**55.2**	38.7	19.5	38.7	49.0
	CL-DETR	Deformable DETR	10%	37.5	55.1	40.4	**20.9**	40.8	50.7
	DiffKA	Deformable DETR	0%	37.2	55.1	39.8	19.4	40.0	50.3
	DiffKA$^+$	Deformable DETR	10%	**37.7**	**55.2**	**40.8**	20.2	**41.3**	**51.0**

Table 1: Comparison results (%) with two-phase setting on COCO 2017 dataset. Buffer rate refers to the proportion of old training samples, which are stored and available in incremental learning phases. DiffKA$^+$ means DiffKA with promotion of 10% buffer.

Method	Upper bound (0-80)	40+20×2		40+10×4			
		+(40-60)	+(60-80)	+(40-50)	+(50-60)	+(60-70)	+(70-80)
RILOD	40.2/58.3	27.8/42.8	15.8/24.0	25.4/38.9	11.2/17.3	10.5/15.6	8.4/12.5
SID		34.0/51.8	23.8/36.5	34.6/52.1	24.1/38.0	14.6/23.0	12.6/23.3
ERD		36.7/54.6	32.4/48.6	36.4/53.9	30.8/46.7	26.2/39.9	20.7/31.8
CL-DETR	43.3/62.4	–	35.3/ –	–	–	–	28.1/ –
DiffKA		**36.8/54.8**	**36.1/54.2**	**36.6/55.5**	**32.7/49.6**	**31.6/47.2**	**29.6/45.8**

Table 2: mAP and AP_{50} results (%) of comparative methods with multi-stage setting on COCO 2017 dataset. '–' denotes the missing values.

4 Experiments

4.1 Experimental Settings

Datasets and Comparisons. We evaluate DiffKA with the widely-used COCO 2017 dataset [Lin *et al.*, 2014]. For comparisons, we adopt six famous IOD methods, i.e., LwF [Li and Hoiem, 2016], RILOD [Li *et al.*, 2019], iCaRL [Rebuffi *et al.*, 2017], ERD [Feng *et al.*, 2022], SID [Peng *et al.*, 2021], and CL-DETR [Liu *et al.*, 2023a]. Results of comparative methods are from [Liu *et al.*, 2023a] and [Junsu *et al.*, 2024].

Metrics. Following [Feng *et al.*, 2022], we use six standard COCO evaluation metrics, i.e., mean Average Precision (mAP), AP50, AP75, AP_S, AP_M, and AP_L. After the first phase of training, we introduce Forgetting Percentage Points (FPP) metric [Liu *et al.*, 2023a] to measure the decline in mAP for the previously learned classes.

Scenario setup. Following [Liu *et al.*, 2023a], we consider two scenarios, i.e., the two-phase and multi-phase setting. In two-phase setting, the model is trained on A class using $\frac{A}{A+B}$ training samples, and the remaining $\frac{B}{A+B}$ samples are used to train the model on the new B class during the incremental learning. In experiments, we define two-phase settings, i.e., $A = 40, B = 40$, and $A = 70, B = 10$, resulting in total training phases $M = 2$, the number of classes in the first learning phase $|\mathcal{C}_1| = A$, and the number of classes in the second learning phase $|\mathcal{C}_2| = B$. In multi-phase setting, the model requires to recognize $P + X \times Y$ classes, where the model is trained with P classes in the first learning phase, and then incrementally learns X new classes in each learning phase. Therefore, we could define $M = Y + 1$, $|\mathcal{C}_1| = P$, and $|\mathcal{C}_2| = \cdots = |\mathcal{C}_M| = X$. In our experiments, we set $40 + 20 \times 2$ and $40 + 10 \times 4$ for multi-phase experiments.

Implementation Details. DiffKA is bulit based on Deformable-DETR [Zhu *et al.*, 2021] with its original settings. In incremental phases, we freeze the coarse Deformable-DETR and initialize a new detector with its parameters to train for 50 epochs in each phase.

4.2 Comparisons with Other Methods

Two-phase Setting. Table 1 presents the experimental results with the two-phase setting, where DiffKA achieves SOTA results on AP_m with both settings, proving its effectiveness in knowledge generation and aggregation across different learning phases. In 70+10 setting, DiffKA outperforms replay-based methods (e.g., CL-DETR) with no access to old-class samples, indicating that DiffKA greatly reduces semantical dependence on old-class samples through knowledge generating and storing in ICT. Meanwhile, DiffKA ranks second in AP_S, where its performance is slightly smaller, i.e., 0.3%, than best performance achieved by CL-DETR. Since DiffKA is built based on Deformable-DETR for knowledge generation, DiffKA is less effective in small object detection.

Multi-phase Setting. Table 2 represents the comparison results with the multi-phase setting, where DiffKA achieves SOTA results with various settings. It's noted that errors accumulates and catastrophic forgetting exacerbates with more phases of learning. However, DiffKA exhibits a more significant improvement, i.e., 1.5 % in mAP, with more chal-

ICT	EUS	CID	AP_{all}	AP_{old}	AP_{new}	FPP
			33.9	33.9	-	9.4
✓			39.5	41.9	22.3	1.4
✓	✓		39.2	41.5	22.0	1.8
✓	✓	✓	**40.3**	**41.7**	**37.0**	**1.6**

Table 3: Ablation study of ICT, enhanced updating scheme(EUS), CID loss on COCO 2017 dataset with 70+10 two-phase setting.

η	λ_1	λ_2	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0.8	0.5	0.5	38.7	56.7	43.2	21.9	42.9	52.2
0.9	0.5	0.5	**40.3**	**57.6**	**44.1**	**22.9**	**44.0**	**53.3**
1	0.5	0.5	40.1	57.3	44.1	22.5	43.8	53.0
0.9	0.5	0.3	40.2	57.6	44.0	22.7	44.2	53.3
0.9	0.5	0.7	40.3	57.5	44.1	23.0	44.1	53.2
0.9	0.75	0.5	40.2	57.7	43.9	23.0	44.2	53.3

Table 4: Results with different values of parameters η in forward diffusion, threshold parameters λ_1, λ_2 in enhanced updating scheme.

Label Strategy	mAP	AP_{50}	AP_{75}
Random&Individual	27.3	42.0	29.5
Fixed&Individual	39.7	57.2	43.6
Joint&Multiple	**40.3**	**57.6**	**44.1**

Table 5: Ablation study of different strategies for class label set stored in ICT.

lenging 40+10x4 settings, comparing with 0.8 % gained with 40+20x2 setting. Such phenomenon proves the effectiveness of DiffKA in mitigating severe catastrophic forgetting, since intrinsic knowledge within ICT greatly alleviates the forgetting with imbalanced, few, or extreme setting of IOD.

4.3 Ablation Study

Table 3 shows the ablation results of ICT, EUS, and CID Loss. Without EUS and CID loss, ICT is still capable to generate intrinsic knowledge for old-class prototypes, maintaining old-class knowledge to deal with IOD task. Such phenomenon indicates class prototypes and inter-class relations plays a critical role in mitigating forgetting. when CID Loss is removed, EUS causes a slight performance drop, since EUS fails in distinguishing and processing confused instances without proper supervision. With the combination of EUS and CID, ICT achieves optimized and gradually knowledge updating, which mitigates forgetting and promotes the learning of new-class prototypes.

Parameter Setting. Table 4 presents results of DiffKA with different parameters η and λ_1, λ_2. $\eta = 0.9$ ensures the best performance of DiffKA with the most reasonable prototype merging rate, forcing DiffKA to focus on the easily confused classes in forward diffusion. Meanwhile, DiffKA is not sensitive to the varying of λ_1 and λ_2 with high robustness.

Strategy for Class Label Set. In Table 5, we evaluate different strategies for class labels set, where Random and Fixed refer to randomly select or select the first individual label from $\mathcal{L}_t$ for supervision respectively, Joint refers to contain multiple labels of all leave node as supervised information. Random strategy fails to capture intrinsic knowledge with a significant performance drop. Fixed strategy achieves worse results than joint strategy, since multiple and joint labels is efficient in describing inter-class associations stored in ICT.

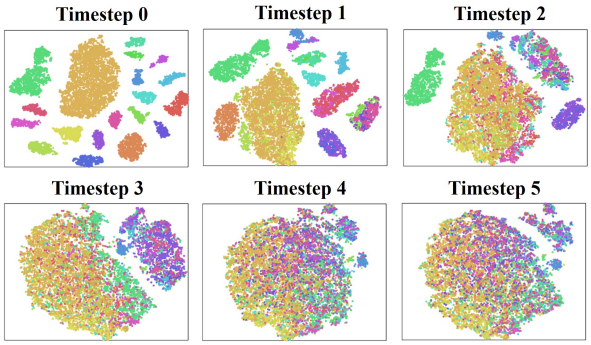

Figure 4: Visualization of semantic space of ICT at each timestep in 70+10 two-phase setting, where each point represents a object instances and different color donates different class. We set $\eta = 0.5$ to control prototype merging rate and only visualize part of classes.

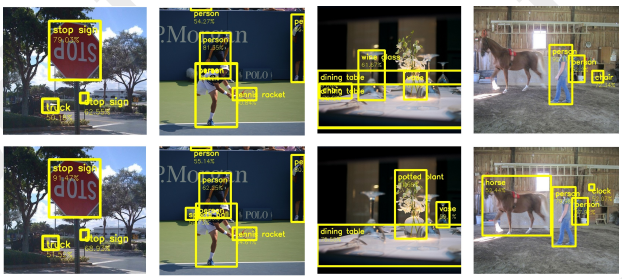

Figure 5: Qualitative results of DiffKA with 70+10 two phase setting, where results of upper row are predicted by Deformable DETR trained with 70 classes, and the bottom row is predicted by DiffKA after incremental learning.

4.4 Qualitative Analysis

In Fig. 4, we show the visualization of semantic space at different forward timesteps, which proves that DiffKA effectively simulates the diffusion process to hierarchically establish inter-class relationships for IOD task. Fig. 5 shows predictions on samples from COCO 2017 dataset. As shown in Fig. 5(a) and (b), DiffKA preserves predictions from previous learning phases, even enhancing the detection confidences of closely related classes in ICT. Several challenging cases containing both old and new-class instances are represented in Fig. 5(c) and (d), where we can find DiffKA accurately predicts the new-class instances (e.g., potted plant and horse), and occasionally struggles with difficult instances, such as blurred or small objects.

5 Conclusion

DiffKA generates and aggregates intrinsic knowledge with forward and backward diffusion, establishing rigid class boundaries and maintaining semantic consistency across training phases and semantic levels.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grants 2023YFC3006501 and 2021YFB3900601, the Natural Science Foundation of Jiangsu Province of China under Grant BK20242050, the Major Science and Technology Program of the Ministry of Water Resources of China under Grant SKS2022072, and the Fundamental Research Funds for the Central Universities under Grants B250201042 and B250201046. It was also supported by the High Performance Computing Platform, Hohai University.

References

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of European Conference on Computer Vision*, volume 12346, pages 213–229, 2020.

[Chen *et al.*, 2023] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 19773–19786, 2023.

[Feng *et al.*, 2022] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9427–9436, 2022.

[Gurbuz *et al.*, 2024] Mustafa Burak Gurbuz, Jean Michael Moorman, and Constantine Dovrolis. NICE: neurogenesis inspired contextual encoding for replay-free class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23659–23669, 2024.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceeding of Advances in Neural Information Processing Systems*, 2020.

[Junsu *et al.*, 2024] Kim Junsu, Cho Hoseong, Kim Jihyeon, Tiruneh Yihalem Yimolal, and Baek Seungryul. Sddgr: Stable diffusion-based deep generative replay for class incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28772–28781, 2024.

[Kang *et al.*, 2023] Mengxue Kang, Jinpeng Zhang, Jinming Zhang, Xiashuang Wang, Yang Chen, Zhe Ma, and Xuhui Huang. Alleviating catastrophic forgetting of incremental object detection via within-class and between-class knowledge distillation. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 18848–18858, 2023.

[Kothapalli, 2023] Vignesh Kothapalli. Neural collapse: A review on modelling principles and generalization. *Trans. Mach. Learn. Res.*, 2023, 2023.

[Li and Hoiem, 2016] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Proceedings of European Conference on Computer Vision*, volume 9908, pages 614–629, 2016.

[Li *et al.*, 2019] Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry P. Heck. RILOD: near real-time incremental learning for object detection at the edge. In *Proceedings of symposium on edge computing*, pages 113–126, 2019.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proceedings of European Conference on Computer Vision*, volume 8693, pages 740–755, 2014.

[Liu *et al.*, 2023a] Yaoyao Liu, Bernt Schiele, Andrea Vedaldi, and Christian Rupprecht. Continual detection transformer for incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23799–23808, 2023.

[Liu *et al.*, 2023b] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost van de Weijer. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 11333–11343, 2023.

[Nichol and Dhariwal, 2021] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171, 2021.

[Peng *et al.*, 2021] Can Peng, Kun Zhao, Sam Maksoud, Meng Li, and Brian C. Lovell. SID: incremental learning for anchor-free object detection via selective and inter-related distillation. *Comput. Vis. Image Underst.*, 210:103229, 2021.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5533–5542, 2017.

[Rezatofighi *et al.*, 2019] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 658–666, 2019.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10674–10685, 2022.

[Wu *et al.*, 2023] Aming Wu, Da Chen, and Cheng Deng. Deep feature deblurring diffusion for detecting out-of-

distribution objects. In *Proceedings of the IEEE/CVF Conference on International Conference on Computer Vision*, pages 13335–13345, 2023.

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