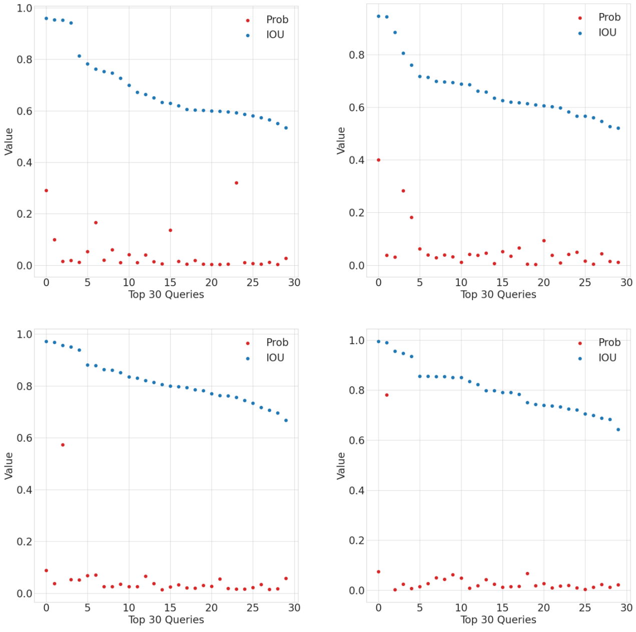
A. More Experiments about Stable Matching

We visualize the queries with top 30 IOU scores of DINO and Stable-DINO in Fig. 6. It shows that Stable-DINO has better alignment between IOU and probability scores.



(a) DINO

(a) Stable-DINO

Figure 6: Comparison of top 30 queries with highest IOU values in DINO (a) and Stable-DINO (b).

B. Details of Memory Fusion

The implementation of memory fusion. The implementation of memory fusion has been depicted in Fig.7. The fusion was executed in a very simple way. The outputs from each encoder layer were amassed and subsequently concatenated with the backbone features along the channel dimension. Following concatenation, a linear projection layer, in conjunction with a norm layer, was employed to project the channel dimension, aligning it to the dimension of the decoder layer. And the fused features were then forwarded to the decoding stage.

How Memory Fusion Works? In the DETR variants, a pre-trained backbone model is often utilized for feature extraction from the input raw images which is typically pre-trained on large-scale dataset such as ImageNet [10]. The extracted features are merged with position encodings and fed into the transformer encoder for extracting and fusing global and local information. While the encoder and backbone can be seen as the same meta-framework for feature extraction but differ in their initialization ways. The encoder's weights are randomly initialized, whereas the backbone features are pre-trained, which means the encoder's feature extraction capability is insufficient in the early stages of training. By fusing the pre-trained backbone features with the multi-scale features processed by the encoder, we enable the decoder to better utilize the pre-trained backbone features during the early training stages. As illustrated in Fig.5, our stable matching strategy significantly accelerates the convergence speed in the early iterations of the training process. Moreover, our newly designed dense memory fusion technique can further boost the convergence speed based on this foundation.

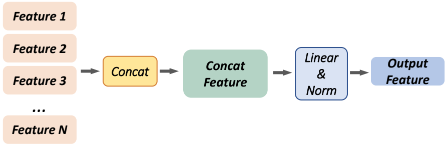


Figure 7: Detailed operation of memory fusion.

C. SOTA experiments

To verify the scalability of our models, we verify our Stable-DINO with large-scale datasets and models. After pre-trained on Objects365 [39], Stable-DINO reaches 63.7 AP on `val2017` and 63.8 AP on `test-dev` without test-time augmentation. We set a new SOTA with under the same setting. The results are available in Table 10.

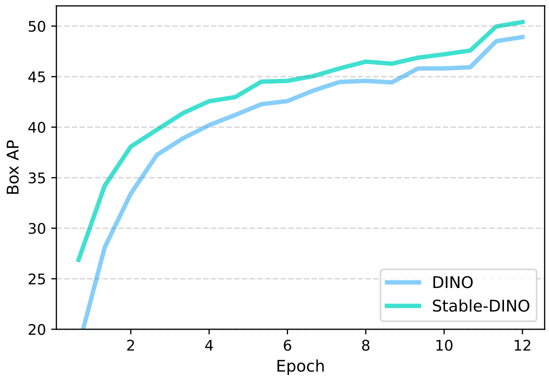


Figure 8: Convergence comparison of DINO and Stable-DINO.

Method	Params	Backbone Pre-training Dataset	Detection Pre-training Dataset	Use Mask	Use TTA	End-to-end	val2017 (AP)	test-dev (AP)
SwinL [30]	284M	IN-22K-14M	O365	✓	✓		58.0	58.7
DyHead [7]	$\geq$ 284M	IN-22K-14M	Unknown*		✓		58.4	60.6
Soft Teacher+SwinL [42]	284M	IN-22K-14M	O365	✓	✓		60.7	61.3
GLIP [24]	$\geq$ 284M	IN-22K-14M	FourODs [24],GoldG+ [24, 20]		✓		60.8	61.5
Florence-CoSwin-H[44]	$\geq$ 637M	FLD-900M [44]	FLD-9M [44]		✓		62.0	62.4
SwinV2-G [29]	3.0B	IN-22K-ext-70M [29]	O365	✓	✓		62.5	63.1
DINO-SwinL	218**M**	IN-22K-14M	O365		✓	✓	**63.2**	**63.3**
Stable-DINO-SwinL(Ours)	**218M**	IN-22K-14M	O365			✓	**63.7**	**63.8**

Table 10: Comparison of the best detection models on MS-COCO. Similar to DETR [3], we use the term “end-to-end” to indicate if a model is free from hand-crafted components like RPN and NMS. The term “TTA” means test-time augmentation. The term “use mask” means whether a model is trained with instance segmentation annotations. We use the terms “IN” and “O365” to denote the ImageNet [11] and Objects365 [39] datasets, respectively. Note that “O365” is a subset of “FourODs” and “FLD-9M”. * DyHead does not disclose the details of the datasets used for model pre-training.

D. Convergence Comparison

We compare the convergence speed of Satble-DINO and DINO in Fig. 8. It shows that Stable-DINO convergence faster than DINO.