

Supplementary Materials: Fractional Correspondence Framework in Detection Transformer

Anonymous Authors

1 IMPLEMENTING SINGLE SHOT MULTIBOX DETECTOR (SSD)

The SSD [2] is an object detection framework designed to predict object bounding boxes and corresponding class probabilities in a single forward pass of the network. It uses a matching cost based on the Intersection over Union (IoU) between the predefined anchor boxes \tilde{b}_i and the ground truth boxes b_j . Specifically, the matching cost is defined as $C_{\text{match}}(\hat{o}_j, o_i^*) = 1 - \text{IoU}(\tilde{b}_i, b_j)$, where a lower cost indicates a better match. The matching process initially pairs each ground truth with the nearest anchor box. Subsequently, anchor boxes are matched to ground truth objects if the IoU surpasses a threshold of 0.5. Within our framework, this initial matching phase is equivalent to setting $\kappa_1 = 0$ and $\kappa_2 = 100$, which focuses on ensuring that all ground truths are matched. In the second phase, the parameters are switched $\kappa_1 = 100$ and $\kappa_2 = 0$, prioritizing that predictions are matched to unique ground truths, akin to enforcing a one-to-one matching as in the Hungarian algorithm. By incorporating entropic regularization, we use the Sinkhorn algorithm to resolve this matching process. It is critical to differentiate the matching cost used during the detection process from the loss function used for training the model. The training loss L_{train} combines a cross-entropy term for classification L_{CE} and a smooth l_1 loss for bounding box regression $L_{\text{smooth } l_1}$, such that $L_{\text{train}}(\hat{o}_j, o_i^*) = \lambda_{CE} L_{CE}(\hat{c}_j, c_i^*) + \lambda_{\text{smooth } l_1} L_{\text{smooth } l_1}(\hat{b}_j, b_i^*)$. This dual-component loss function ensures that the model is trained to both correctly classify objects and accurately predict their bounding box locations. We set $\lambda_{\text{smooth } l_1} = 1$, $\lambda_{l_1} = 5$, and $\lambda_{CE} = 1$.

2 DISCUSSION ON TRANSPORT PLANS

Optimal transport involves the task of minimizing transportation costs between two distributions (prediction and ground truth objects) $\nu \in P_+(X)$ and $\mu \in P_+(Y)$ with equal mass, i.e., $\int_X \nu dx = \int_Y \mu dy$. This is achieved using a cost function $c : X \times Y \rightarrow [0, +\infty]$, formalized as:

$$\inf \left\{ \int_{X \times Y} c(x, y) d\Gamma(x, y) : \Gamma \in U(\nu, \mu) \right\},$$

where $U(\nu, \mu)$ denotes the set of possible transport plans,

$$U(\nu, \mu) = \left\{ \Gamma \in P_+(X \times Y) : \int_Y d\Gamma = \nu \text{ and } \int_X d\Gamma = \mu \right\}.$$

The optimal solution is termed the optimal transport plan Γ . In our framework, the probability simplex Δ^N is replaced by the space of probability distributions $P_+(X)$ on X . The transport plans are defined as the set of joint probability distributions $\Gamma \in P_+(X \times Y)$, whose marginal distributions are ν and μ , and cost function $c = C_{\text{match}}$. This optimization defines the Wasserstein distance between ν and μ , as $W_p(\nu, \mu)$, assuming the cost function c represents a distance $c = d^p$ for some exponent $p \geq 1$ [4]. However, our matching cost, C_{match} does not meet the criteria necessary for discussing a

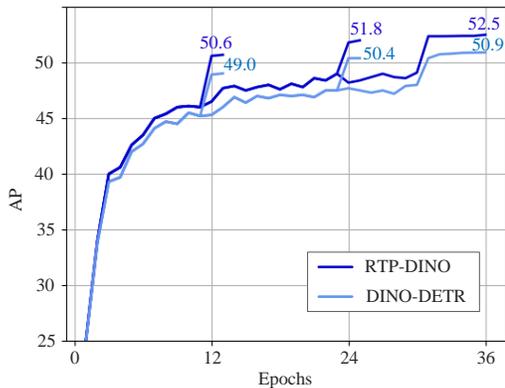


Figure 1: Training convergence curves evaluated on COCO val2017 for the recent state-of-the-art DINO-DETR and our RTP-DINO with ResNet-50 backbone. The x-axis denotes the number of epochs, and the y-axis indicates mAP. Our model not only achieving a better AP but also by doing so in a shorter span of training epochs, thus demonstrating an improved learning efficiency over DINO. The key to our model fast convergence is the integration of entropy regularization – a feature absent in the DINO model. Entropy regularization facilitates a smoother and more stable optimization, allowing for faster adaptation during training iterations.

Wasserstein distance, as it lacks properties like triangular inequality and symmetry. Thus, we cannot discuss the Wasserstein distance in this context [1, 3]. Indeed, triangular inequality ensures that the direct route between two points is always the shortest. In other words, for any three points a , b , and c , the cost of going from a to c should hold the inequality $d(a, c) \leq d(a, b) + d(b, c)$ [3]. C_{match} does not satisfy this condition because the sum of individual distances between predictions and ground truths may not accurately reflect the collective distance. Additionally, symmetric means that the cost from point a to point b should be the same as from point b to point a , i.e., $d(a, b) = d(b, a)$. Matching costs derived from functions like cross-entropy are asymmetric as they depend on the order of the arguments (predicted vs. actual labels), reflecting a directed discrepancy rather than a mutual distance. This aspect has been discussed more extensively in [5].

3 EXPERIMENT RESULTS

Figure 1 shows the training convergence of our RTP-DETR model alongside DINO-DETR. Our model not only improves average precision (AP) but also reaches these improvements faster, within a few epochs. In Figure 2, we investigate the adaptability of our matching strategy under the influence of different parameter settings.

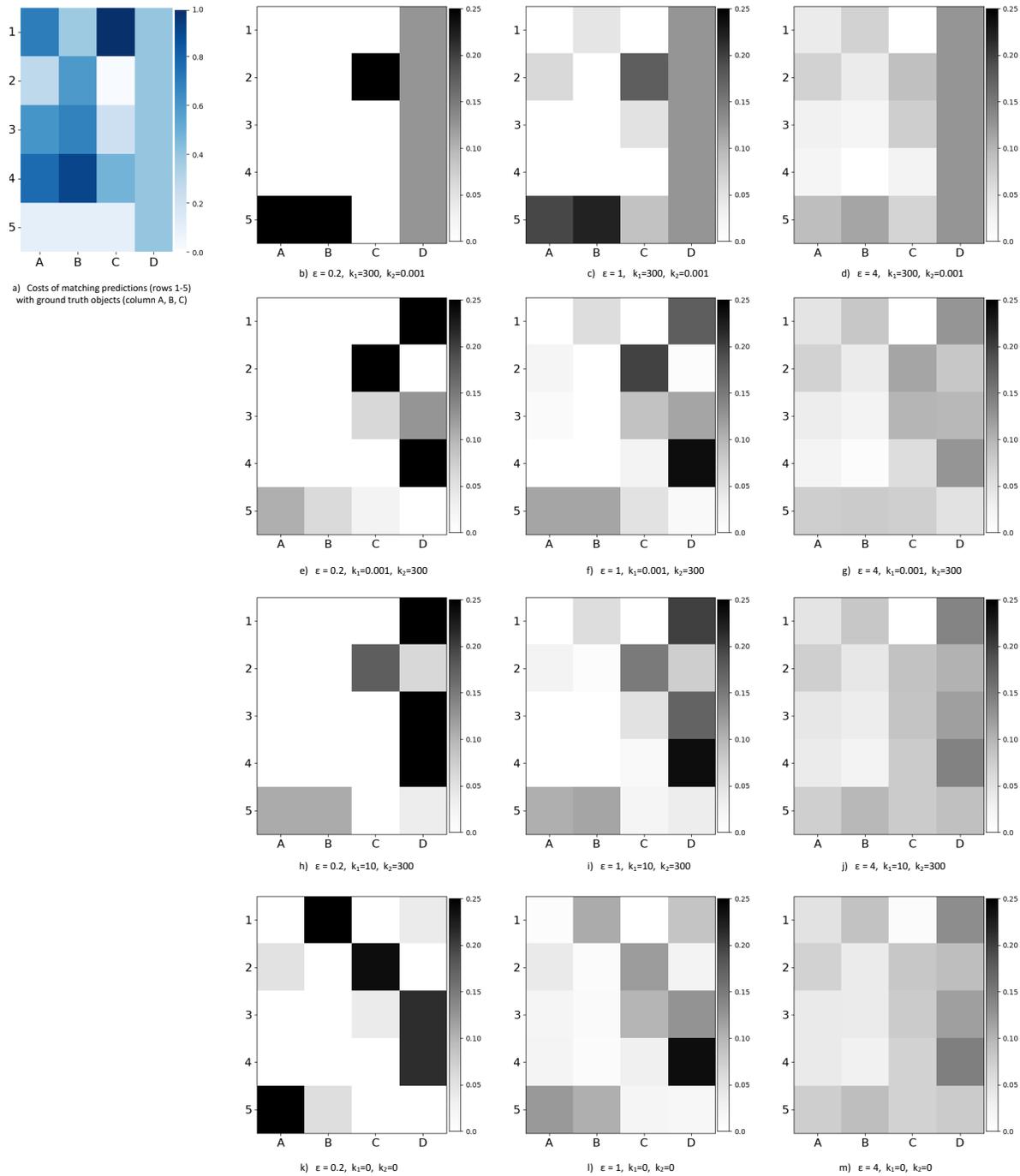


Figure 2: Different configuration of our model based on varying parameter values. (a) presents the matching cost, where each prediction is denoted by a number and each ground truth object by a letter. The costs are represented using a color gradient, darker colors indicate higher costs, and lighter colors, approaching to white, signify lower costs (better match). The background cost (column D) is set at 0.48, penalizes predictions that do not match any ground truth objects (acting as a threshold for determining mismatches). Subfigures (b)-(m) illustrate the actual matching outcomes where a black square represents a full match between a prediction and a ground truth ($\Gamma_{ij} = 1$), and a white square indicates no match.

The figure illustrates how different parameter values lead to distinct matching outcomes, each providing insight into the complex relationships between predictions and ground truth objects.

REFERENCES

- [1] Lenaic Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. 2020. Faster Wasserstein distance estimation with the Sinkhorn divergence. *Advances in Neural Information Processing Systems* 33 (2020), 2257–2269.
- [2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*. 21–37.
- [3] Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning* 11, 5-6 (2019), 355–607.
- [4] Cédric Villani. 2021. *Topics in optimal transportation*. Vol. 58. American Mathematical Soc.
- [5] Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. 2021. Asymmetric loss functions for learning with noisy labels. In *International conference on machine learning*. PMLR, 12846–12856.

233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348