GoMatching: A Simple Baseline for Video Text Spotting via Long and Short Term Matching

Haibin He[1*], **Maoyuan Ye**[1*], **Jing Zhang**[1†], **Juhua Liu**[1†], **Bo Du**[1], **Dacheng Tao**[2]
[1] School of Computer Science, National Engineering Research Center for Multimedia Software, and Institute of Artificial Intelligence, Wuhan University, China
[2] College of Computing & Data Science at Nanyang Technological University
{haibinhe, yemaoyuan, liujuhua, dubo}@whu.edu.com
jingzhang.cv@gmail.com, dacheng.tao@gmail.com
https://github.com/Hxyz-123/GoMatching

Abstract

Beyond the text detection and recognition tasks in image text spotting, video text spotting presents an augmented challenge with the inclusion of tracking. While advanced end-to-end trainable methods have shown commendable performance, the pursuit of multi-task optimization may pose the risk of producing sub-optimal outcomes for individual tasks. In this paper, we identify a main bottleneck in the state-of-the-art video text spotter: the limited recognition capability. In response to this issue, we propose to efficiently turn an off-the-shelf query-based image text spotter into a specialist on video and present a simple baseline termed GoMatching, which focuses the training efforts on tracking while maintaining strong recognition performance. To adapt the image text spotter to video datasets, we add a rescoring head to rescore each detected instance's confidence via efficient tuning, leading to a better tracking candidate pool. Additionally, we design a long-short term matching module, termed LST-Matcher, to enhance the spotter's tracking capability by integrating both long- and short-term matching results via Transformer. Based on the above simple designs, GoMatching delivers new records on ICDAR15-video, DSText, BOVText, and our proposed novel test set with arbitrary-shaped text termed ArTVideo, which demonstrates GoMatching's capability to accommodate general, dense, small, arbitrary-shaped, Chinese and English text scenarios while saving considerable training budgets.

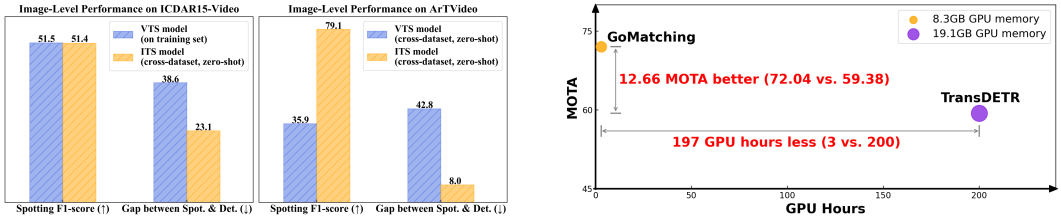
1 Introduction

Text spotting has received increasing attention due to its various applications, such as video retrieval [1] and autonomous driving [2]. Recently, numerous image text spotting (ITS) methods [3–6] that simultaneously tackle text detection and recognition, have attained extraordinary accomplishment. In the video realm, video text spotting (VTS) involves a tracking task additionally. Although VTS methods [7–12] make significant progress, a substantial discrepancy persists when compared to ITS. We observe that the text recognition proficiency of VTS models is far inferior to ITS models. To investigate this issue, we compare the state-of-the-art (SOTA) VTS model TransDETR [12] and ITS model Deepsolo [5] for image-level text spotting performance on ICDAR15-video [13] and our established ArTVideo (*i.e.*, **Ar**bitrary-shaped **T**ext in **Video**) test set (Sec.4.1) which comprises about 30% curved text.

As illustrated in Fig. 1(a), even when evaluating the image-level spotting performance on the VTS model's training set, the F1-score of TransDETR is only comparable to the zero-shot performance of

[*]Equal contribution, † Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) (b)

Figure 1: (a) 'Gap between Spot. & Det.': the gap between spotting and detection F1-score. As the spotting task involves recognizing the results of the detection process, the detection score is indeed the upper bound of spotting performance. The larger the gap, the poorer the recognition ability. Compared to the ITS model (Deepsolo [5]), the VTS model (TransDETR [12]) presents unsatisfactory image-level text spotting F1-scores, which lag far behind its detection performance, especially on ArTVideo with curved text. It indicates recognition capability is a main bottleneck in the VTS model. (b) GoMatching outperforms TransDETR by over 12 MOTA on ICDAR15-video while saving 197 training GPU hours and 10.8GB memory. Notice that since the pre-training strategies and settings vary between TransDETR and GoMatching, the comparison is focused on the fine-tuning stage.

Deepsolo. The performance of the VTS model on ArTVideo is much worse. Moreover, there is a huge gap between the spotting and detection-only performance of the VTS model, which indicates that the recognition capability is the main bottleneck. We attribute this discrepancy to two key aspects: 1) the model architecture and 2) the training data. First, in terms of model architecture, ITS studies [5, 6] have presented the advantages of employing advanced query formulation for text spotting in DETR frameworks [14, 15]. In contrast, existing Transformer-based VTS models still rely on Region of Interest (RoI) components or simply cropping detected text regions for recognition. On the other hand, some studies [16, 17] have indicated that there exists optimization conflict in detection and association during the end-to-end training of MOTR [18]. We hold that TransDETR [12], which further incorporates text recognition into MOTR-based architecture, may also suffer from optimization conflict. Second, regarding the training data, most text instances in current video datasets [13, 10, 19] are straight or oriented, and the bounding box labels are only quadrilateral, which constrains the data diversity and recognition performance as well. Overall, the limitations in model architecture and data probably lead to the unsatisfactory text spotting performance of the SOTA VTS model. Hence, ***leveraging model and data knowledge from ITS presents considerable value for VTS***.

To achieve this, a straightforward approach is to take an off-the-shelf SOTA image text spotter and focus the training efforts on tracking across frames, akin to tracking-by-detection methods. An important question is how to efficiently incorporate a RoI-free image text spotter for VTS. In this paper, we propose a simple baseline via lon**G** and sh**o**rt term **Matching**, termed **GoMatching**, which leverages an off-the-shelf RoI-free image text spotter to identify text from each single frame and associates text instances across frames with a strong tracker.

Specifically, we select the state-of-the-art DeepSolo [5] as the image text spotter and design a **L**ong-**S**hort **T**erm Matching-based tracker termed **LST-Matcher**. Initially, to adapt the DeepSolo to video datasets while preserving its inherent knowledge, we freeze Deepsolo and introduce a rescoring mechanism. This mechanism entails training an additional lightweight text classifier called rescoring head via efficient tuning, and recalibrating confidence scores for detected instances to mitigate performance degradation caused by the image-video domain gap. The final score for each instance is determined by a fusion operation between the original score provided by the image text spotter and the calibrated score acquired from the rescoring head. The identified text instances are then sent to LST-Matcher for association. LST-Matcher can effectively harnesses both long- and short-term information, making it a highly capable tracker. As a result, our baseline significantly surpasses existing SOTA methods by a large margin with much lower training costs, as shown in Fig. 1(b).

In summary, the contribution of this paper is threefold. **1)** We identify the limitations in current VTS methods and propose a novel and simple baseline, which leverages an off-the-shelf image text spotter with a strong customized tracker. **2)** We introduce the rescoring mechanism and long-short term matching module to adapt image text spotter to video datasets and enhance the tracker's capabilities. **3)** We establish the ArTVideo test set for addressing the absence of curved texts in current video datasets and evaluating the text spotters on videos with arbitrary-shape text. Extensive

experiments on public challenging datasets and the established ArTVideo test set demonstrate the effectiveness of our baseline and its outstanding performance with less training budgets. For example, GoMatching achieves the highest ranking on ICDAR15-video and DSText. Especially on bilingual dataset BOVText, GoMatching obtains a 45% improvement on MOTA compared to the recorded best performance [11]. On curved text dataset ArTVideo, GoMatching also surpasses previous SOTA method [12] by a substantial margin.

2 Related Works

2.1 Multi-Object Tracking

Multi-object tracking methods follow the tracking-by-detection (TBD) or tracking-by-query-propagation (TBQP) pipeline. TBD methods [20–22] employ detectors for localization and then use association algorithms to get object trajectories. Different from extending tracks frame-by-frame, GTR [23] proposes to generate entire trajectories at once in Transformer. TBQP paradigm extends query-based object detectors[14, 15] to tracking. MOTR [18] detects object locations and serially updates its tracking queries for detecting the same items in the following frames, achieving an end-to-end solution. However, MOTR suffers from optimization conflict between detection and association [16, 17], resulting in inferior detection performance. For the VTS task which additionally involves text recognition, a naive way of training all modules end-to-end may also lead to optimization conflict. In contrast, we explore inheriting prior knowledge of text spotting from ITS models while focusing on the tracking task.

2.2 Image Text Spotting

Early approaches [24–26] crafted RoI-based modules to bridge text detection and recognition. However, these methods ignored one vital issue, *i.e.*, the synergy problem between the two tasks. To overcome this dilemma, recent Transformer-based methods [27, 3, 28, 6] get rid of the fetters of RoI modules, and chase a better representation for the two tasks. For example, DETR-based TESTR [4] uses two decoders for each task in parallel. In contrast, DeepSolo [5] proposes a unified and explicit query form for the two tasks, without harnessing dual decoders. However, the above methods cannot perform tracking in the video.

2.3 Video Text Spotting

Compared to ITS, existing SOTA VTS methods still rely on RoI for recognition. CoText [11] adopts a lightweight text spotter with Masked-RoI, then uses several encoders to fuse features derived from the spotter, and finally feeds them to a tracking head with cosine similarity matching. TransDETR [12] performs detection and tracking under the MOTR paradigm and then uses Rotated-RoI to extract features for the subsequent recognizer. They pursue training all modules in an end-to-end manner. In comparison, we explore how to efficiently turn a RoI-free ITS model into a VTS one. We reveal the probability of freezing off-the-shelf ITS part and focusing on tracking, thereby saving training budgets while reaching SOTA performance.

3 Methodology

3.1 Overview

The architecture of GoMatching is presented in Fig. 2. It consists of a frozen image text spotter, a rescoring head, and a Long-Short Term Matching module (LST-Matcher). We adopt an outstanding off-the-shelf image text spotter (*i.e.*, DeepSolo) and freeze its parameters, with the aim of introducing strong text spotting capability into VTS while significantly reducing training cost. In DeepSolo, there are p sequences of queries used for final predictions, with each storing comprehensive semantics for a text instance. To alleviate spotting performance degradation caused by the image-video domain gap, we devise a rescoring mechanism, which determines the confidence scores for text instances by considering both the scores from the image text spotter and a new trainable rescoring head. Finally, we design LST-Matcher to generate instance trajectories by leveraging long-short term information.

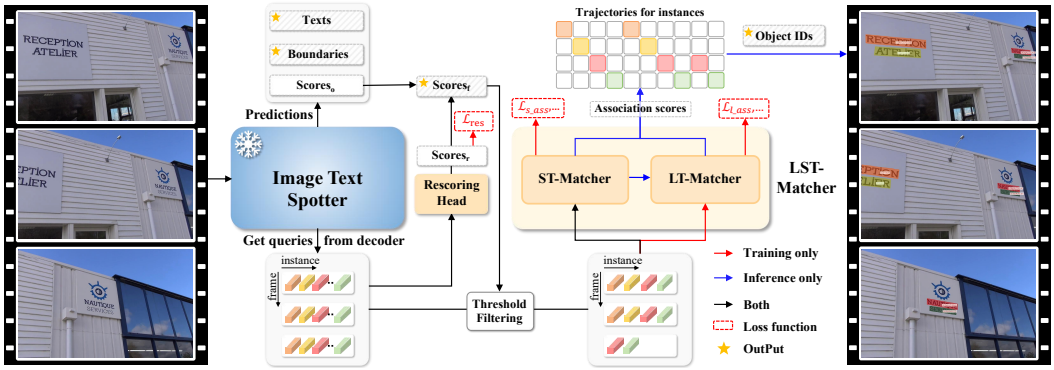


Figure 2: **The overall architecture of GoMatching**. The frozen image text spotter provides text spotting results for frames. The rescoring mechanism considers both instance scores from the image text spotter and a trainable rescoring head to reduce performance degradation due to the domain gap. Long-short term matching module (LST-Matcher) assigns IDs to text instances based on the queries in long-short term frames. The yellow star sign '★' indicates the final output of GoMatching.

3.2 Rescoring Mechanism

Owing to the domain gap between image and video datasets, employing a frozen image text spotter for direct prediction may result in relative low recall due to low text confidence, further leading to a reduction in end-to-end spotting performance. To ease this issue, we devise a rescoring mechanism via a lightweight rescoring head and a simple score fusion operation. Specifically, the rescoring head is designed to recompute the score for each query from the decoder in the image text spotter. It consists of a simple linear layer and is initialized with the parameters of the image text spotter's classification head. The score fusion operation then decides the final scores by considering both the scores from the image text spotter and the rescoring head. Let $C_o^t = \{c_{o_1}^t, ..., c_{o_p}^t\}$ be a set of original scores produced by image text spotter in frame t. $C_r^t = \{c_{r_1}^t, ..., c_{r_p}^t\}$ is a set of recomputed scores obtained from the rescoring head. We obtain the maximum value for each query as the final score, denoted as $C_f^t = \{c_{f_1}^t = max(c_{o_1}^t, c_{r_1}^t), ..., c_{f_p}^t = max(c_{o_p}^t, c_{r_p}^t)\}$. With final scores, the queries in frames are filtered by a threshold before being sent to LST-Matcher for association.

3.3 Long-Short Term Matching Module

Long-short term matching module (LST-Matcher) consists of two sub-modules: the Short Term Matching module (ST-Matcher) and the Long Term Matching module (LT-Matcher), which own the same structure. ST-Matcher is steered to match simple instances between adjacent frames into trajectories, while LT-Matcher is responsible for using long term information to address the unmatched instances due to severe occlusions or strong appearance changes. Each of them contains a one-layer Transformer encoder and a one-layer Transformer decoder [23]. We use a simple multi-layer perceptron (MLP) to map the filtered text instance queries into embeddings as the input, getting rid of using RoI features as in most existing MOT methods. In the encoder, historical embeddings are enhanced by self-attention. The decoder takes embeddings in the current frame as query and enhanced historical embeddings as key for cross-attention, and computes the association score matrix. The current instances are then linked to the existing trajectories composed of historical embeddings or generate new trajectories according to the association score matrix.

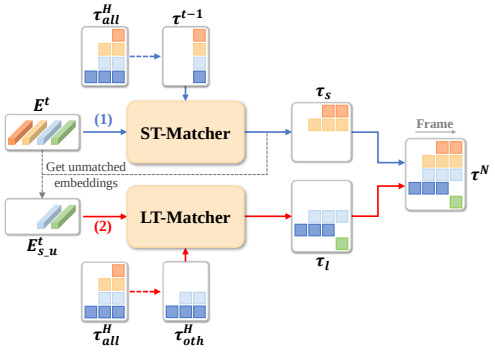


Figure 3: The inference pipeline of LST-Matcher, which is a two-stage association process: (1) ST-Matcher associates the instances with trajectories in previous frames as denoted by blue lines. (2) LT-Matcher associates the remaining unmatched instances by utilizing other trajectories in history frames as denoted by red lines.

To be specific, supposing a given clip including T frames and N_t text instances in frame t after threshold filtering. $Q^t = \{q_1^t, ..., q_{N_t}^t\}$ is the set of text instance queries in frame t. Initially, we use a two-layer MLP to map these frozen queries into embeddings $E^t = \{e_1^t, ..., e_{N_t}^t\}$. The set of embeddings in all frames is denoted as $E^L = E^1 \cup ... \cup E^T$. Let the universal set of embeddings in adjacent frames of the input batch be denoted as $E^S = E^{S_2} \cup E^{S_3} \cup ... \cup E^{S_T}$ and $E^{S_t} = E^{t-1} \cup E^t$. Based on the predictions of image text spotter, we obtain their corresponding bounding boxes $B^t = \{b_1^t, ..., b_{N_t}^t\}$. Let $\tau = \{\tau_1, ..., \tau_K\}$ be the set of ground-truth (GT) trajectories of all instances in the clip, where $\tau_k = \{\tau_k^1, ..., \tau_k^T\}$ describes a tube of instance locations $\tau_k^t \in \mathbb{R}^4 \cup \{\emptyset\}$ through time. $\tau_k^t = \emptyset$ means the absence of instance k in frame t. Let $\hat{\alpha}_k^t$ be the matched instance index for τ_k^t according to the following equation:

$$\hat{\alpha}_k^t = \begin{cases} \emptyset, & \text{if } \tau_k^t = \emptyset \text{ or } \max_i(IoU(b_i^t, \tau_k^t)) < 0.5 \\ \text{argmax}_i(IoU(b_i^t, \tau_k^t)), & \text{otherwise} \end{cases}. \tag{1}$$

ST-Matcher calculates a short-term association score $v_i^t(e_{\hat{\alpha}_k^t}^t, E^{S_t}) \in \mathbb{R}^{N_{S_t}}$ for i-th instances in frame t, where $e_{\hat{\alpha}_k^t}^t \in \mathbb{R}^D$ is a trajectory query and $N_{S_t} = N_t + N_{t-1}$. LT-Matcher calculates a long-term trajectory-specific association score $u_i^t(e_k, E^L) \in \mathbb{R}^N$ for i-th instances in frame t, where $e_k \in \{e_{\hat{\alpha}_k^1}^1, e_{\hat{\alpha}_k^2}^2, ..., e_{\hat{\alpha}_k^T}^T\}, N = \sum_{t=1}^T N_t$. Specifically, when $v_i^t(e_{\hat{\alpha}_k^t}^t, E^{S_t}) = 0$ and $u_i^t(e_k, E^L) = 0$, it means no association at time t. Then, ST-Matcher and LT-Matcher can predict distributions of short-term and long-term associations for all instance i in frame t which can be written as:

$$P_{s_a}(e_{\hat{\alpha}_k^t}^t, E^{S_t}) = \frac{\exp(v_i^t(e_{\hat{\alpha}_k^t}^t, E^{S_t}))}{\sum_{j \in \{\emptyset, 1, ..., N_t\}} \exp(v_j^t(e_{\hat{\alpha}_k^t}^t, E^{S_t}))}, \tag{2}$$

$$P_{l_a}(e_k, E^L) = \frac{\exp(u_i^t(e_k, E^L))}{\sum_{j \in \{\emptyset, 1, ..., N_t\}} \exp(u_j^t(e_k, E^L))}. \tag{3}$$

To ensure sufficient training of ST-Matcher and LT-Matcher, embeddings set E^S and E^L are fed into ST-Matcher and LT-Matcher during training, respectively.

During inference, we engage a memory bank to store the instance trajectories from H history frames for long term association. All filtered instances in each frame are further processed by non-maximum-suppression (NMS) before being fed into LST-Matcher for association. Unlike the training phase, where ST-Matcher and LT-Matcheder are independent of each other, LST-Matcher comprises a two-stage associating procedure as described in Fig. 3. Concretely, ST-Matcher first matches the embedding E^t in the current frame t with the trajectories τ^{t-1} in the previous frame $t-1$. Then, LT-Matcher employs other trajectories τ_{oth}^H in the memory bank to associate the unmatched ones $E_{s\_u}^t$ with low association score in ST-Matcher caused by the heavy occlusion or strong appearance changes. If the association score with any trajectory calculated in ST-Matcher or LT-Matcher is higher than a threshold θ, the instance is linked to the trajectory with the highest score. Otherwise, this instance is used to initiate a new trajectory. Finally, we combine the trajectories τ_s and τ_l predicted by ST-Matcher and LT-Matcher to obtain new trajectories τ^N for tracking in the next frame.

3.4 Optimization

Rescoring Loss. To train the rescoring head, we following DETR [14] and use Hungarian algorithm [29] to find a bipartite matching $\hat{\sigma}$ between the prediction set $\hat{Y}$ and the ground truth set Y with minimum matching cost $\mathcal{C}$:

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i}^{N} \mathcal{C}(Y_i, \hat{Y}_{\sigma(i)}), \tag{4}$$

where N is the number of ground truth instances per frame. The cost $\mathcal{C}$ can be defined as:

$$\mathcal{C}(Y_i, \hat{Y}_{\sigma(i)}) = \lambda_c \mathcal{L}_{cls}(\hat{p}_{\sigma(i)}(c_i)) + \lambda_b \sum_{1}^{N} \|b_i - \hat{b}_{\sigma(i)}\|, \tag{5}$$

where λ_c and λ_b serve as the hyper-parameters to balance different tasks. $\hat{p}_{\sigma(i)}(c_i)$ and $\hat{b}_{\sigma(i)}$ are the probability for ground truth class c_i and the predicition of bounding box respectively, and b_i

represents the ground truth bounding box. $\mathcal{L}_{cls}$ is the focal loss [30]. Specifically, the focal loss for training the rescoring head can be formulated as:

$$\mathcal{L}_{res} = \sum_{1}^{N}[-\mathbb{1}_{\{c_i \neq \varnothing\}}\alpha(1-\hat{p}_{\hat{\sigma}(i)}(c_i))^{\gamma}\log(\hat{p}_{\hat{\sigma}(i)}(c_i)) - \mathbb{1}_{\{c_i=\varnothing\}}(1-\alpha)(\hat{p}_{\hat{\sigma}(i)}(c_i))^{\gamma}\log(1-\hat{p}_{\hat{\sigma}(i)}(c_i))], \tag{6}$$

where α and γ are the hyper-parameters of focal loss.

Long-Short Association Loss. In ST-Matcher, we only consider each trajectory in the universal set of adjacent frames, while in LT-Matcher we consider each trajectory in all long term frames. For each trajectory, we optimize the log-likelihood of its assignments $\hat{\alpha}_k$ following GTR [23]:

$$\mathcal{L}_{s\_ass}(E^S, \hat{\tau}_k) = -\sum_{t=2}^{T}\log P_{s_a}(\hat{\alpha}_k^t | e^t_{\hat{\alpha}_k^t}, E^{S_t}), \tag{7}$$

$$\mathcal{L}_{l\_ass}(E^L, \hat{\tau}_k) = -\sum_{w}\sum_{t=1}^{T}\log P_{l_a}(\hat{\alpha}_k^t | E^w_{\hat{\alpha}_k^w}, E^L), \tag{8}$$

where $w \in \{1, ..., T | \hat{\alpha}_k^w \neq \emptyset\}$.

In ST-Matcher and LT-Matcher, empty trajectories would be generated for these unassociated queries, and their optimization goals can be defined as:

$$\mathcal{L}_{s\_bg}(E^S) = -\sum_{j:\nexists\hat{\alpha}_k^t=j}\sum_{t=2}^{T}\log P_{s_a}(\alpha^t = \emptyset | e_j^t, E^{S_t}), \tag{9}$$

$$\mathcal{L}_{l\_bg}(E^L) = -\sum_{w=1}^{T}\sum_{j:\nexists\hat{\alpha}_k^w=j}\sum_{t=1}^{T}\log P_{l_a}(\alpha^t = \emptyset | E_j^w, E^L). \tag{10}$$

Finally, we can get the long-short association loss as follows:

$$\mathcal{L}_{asso} = \mathcal{L}_{s\_bg} + \mathcal{L}_{l\_bg} + \sum_{\hat{\tau}_k}(\mathcal{L}_{s\_ass} + \mathcal{L}_{l\_ass}). \tag{11}$$

Overall Loss. Combined with the rescore loss $\mathcal{L}_{res}$ in Eq. (6) and the long-short association loss $\mathcal{L}_{asso}$ in Eq. (11), the final training loss can be defined as:

$$\mathcal{L} = \lambda_{res}\mathcal{L}_{res} + \lambda_{asso}\mathcal{L}_{asso}, \tag{12}$$

where the hyper-parameters λ_{res} and λ_{asso} are the weights of $\mathcal{L}_{res}$ and $\mathcal{L}_{asso}$, respectively.

4 Experiments

4.1 Datasets and Evaluation Metrics

ICDAR15-video [13] is a word-level video text reading benchmark annotated with quadrilateral bounding boxes, comprising a training set of 25 videos and a test set of 24 videos. It focuses on wild scenarios, such as driving on the road, exploring shopping streets, walking in a supermarket, *etc.*

BOVText [10] is a large-scale, bilingual, and open-world benchmark for video text spotting, encompassing English and Chinese. The dataset is meticulously collected from *YouTube* and *KuaiShou* with different scenarios. The text box annotations are represented as quadrilaterals at the textline level.

DSText [19] is a newly proposed dataset, and focuses on dense and small text reading challenges in the video. This dataset provides 50 training videos and 50 test videos. Compared with the previous datasets, DSText mainly includes the following three new challenges: dense video texts, high-proportioned small texts, and various new scenarios, *e.g.*, 'Game', 'Sports', *etc.* Similar to ICDAR15-video, DSText adopts word-level annotations, which are labeled with quadrilaterals.

ArTVideo is a novel word-level test set established in this work to evaluate the performance of arbitrary-shaped video text, which is absent in the VTS community. It contains 20 videos with about 30% curved text instances. Straight text is annotated with quadrilaterals, while curved text is annotated with polygons. More details are provided in the Appendix A.

Evaluation Metrics. To evaluate performance, we adopt three evaluation metrics commonly used in ICDAR15-video competition and DSText competition, including MOTA [31], MOTP, and IDF1 [32].

4.2 Implementation Details

In all experiments, we only use a single NVIDIA GeForce RTX 3090 (24G) GPU to train and evaluate GoMatching. As for the image text spotter in GoMatching, we apply the officially released DeepSolo [5]. During fine-tuning GoMatching on downstream video datasets, we only update the rescoring head and LST-Matcher, while keeping DeepSolo frozen. More inference settings can be seen in the Appendix B.

Training Setting. The text spotting part of GoMatching is initialized with off-the-shelf DeepSolo weights and kept frozen in all experiments. We optimize other modules on video datasets. We follow EfficientDet [33] to adopt the scale-and-crop augmentation strategy with a resolution of 1280. The batch size T is 6. All frames in a batch are from the same video. Text instances with fusion scores higher than 0.3 are fed into the LST-Matcher during training. AdamW [34] is used as the optimizer. We adopt the warmup cosine annealing learning rate strategy with the initial learning rate being set to 0.00005. The loss weights λ_{res} and λ_{asso} are set to 1.0 and 0.5, respectively. For focal loss, α is 0.25 and γ is 2.0 as in [14, 5]. The model is trained for 30k iterations on all downstream video datasets.

4.3 Comparison with State-of-the-art Methods

Results on ICDAR15-video. To evaluate the effectiveness of GoMatching on oriented video text, we conduct a comparison with the state-of-the-art methods on ICDAR15-video presented in Table 1a. As can be seen, GoMatching ranks first in all metrics on the ICDAR15-video leaderboard. By effectively combining a robust image text spotter with a strong tracker, GoMatching improves the best performance by 5.08% MOTA, 0.75% MOTP, and 3.16% IDF1, respectively. Furthermore, owing to the substantial enhancement in recognition and tracking capabilities (details can be found in Sec. 4.4 and Appendix G), GoMatching outperforms the current SOTA single-model method TransDETR by 11.08% MOTA, 3.92% MOTP, and 7.31% IDF1, respectively.

Results on BOVText. Except for the English word recognition scenario, GoMatching can readily adapt to other video text recognition scenarios, such as Chinese text line recognition. For BOVText, which focuses on English and Chinese textline recognition, we employ the DeepSolo trained on bilingual textline datasets and then fine-tune GoMatching on BOVText. The results are presented in Table 1b. It is evident that GoMatching achieves a new record on the BOVText dataset and surpasses previous methods significantly. GoMatching exhibits superior performance over the previous SOTA method CoText [11], with improvements of 41.5% on MOTA, 6.9% on MOTP, and 14.3% on IDF1. Such exceptional performance of GoMatching on BOVText suggests its proficiency in spotting both Chinese and English text in videos. Moreover, it can be easily extended to other languages by adapting the image spotter.

Results on DSText. We further conduct experiments on DSText with dense and small video text scenarios. Results are presented in Table 1c. It is worth noting that most of the previous methods on the DSText leaderboard used an ensemble of multiple models and large public datasets to enhance their performance [19]. For example, *TencentOCR* integrates the detection results of DBNet [35] and Cascade MaskRCNN [36] built with multiple backbone architectures, combines the Parseq [37] text recognizer, and further improves the end-to-end tracking with ByteTrack [22]. *DA* adopts Mask R-CNN [38] and DBNet to detect text, then uses BotSORT [21] to replace the tracker in VideoTextSCM [39] and employs the Parseq model for recognition. As a single model with a frozen image text spotter, GoMatching also shows competitive performance compared to other ensembling methods on the leaderboard. GoMatching ranks first (22.83%) on MOTA, second (80.43%) on MOTP, and third (46.09%) on IDF1. Moreover, compared to the SOTA single-model method, GoMatching achieves substantial improvements of 45.46% and 19.66% on MOTA and IDF1, respectively.

Results on ArTVideo. We test TransDETR and GoMatching on ArTVideo to compare the zero-shot text spotting capabilities for arbitrary-shaped text. For a fair comparison, both TransDETR and GoMatching are trained on ICDAR15-video. Unlike ICDAR15-video and DSText which only have straight text, ArTVideo has a substantial number of curved text, so we report results under four settings: tracking results on both straight and curved text, spotting results on both straight and curved text, tracking results on curved text only, and spotting results on curved text only. As shown in Table 1d, GoMatching outperforms TransDETR under all settings. Especially when involving an additional recognition task (end-to-end spotting) or only considering curved text, the performance advantages of GoMatching are more significant. This further confirms that the previous SOTA

Table 1: **Comparison results with SOTA methods on four distinct datasets.** '†' denotes that the results are collected from the official competition website. '*': we use the officially released model for evaluation. 'M-ME' indicates whether multi-model ensembling is used. 'Y' and 'N' stand for yes and no. The best and second-best results are marked in **bold** and underlined, respectively.

(a) **Results on ICDAR15-video.**

Method	MOTA (↑)	MOTP (↑)	IDF1 (↑)
HIK_OCR [9]	52.98	74.88	61.85
CoText [11]	58.94	74.53	71.66
TransDETR [12]	60.96	74.61	72.80
h&h_lab†	63.76	77.78	71.08
GOCR Offline†	63.05	74.31	76.95
CoText(Kuaishou_MMU)†	66.96	76.55	74.24
GoMatching (size:800) (ours)	68.51	77.52	76.59
GoMatching (size:1000) (ours)	**72.04**	**78.53**	**80.11**
GoMatching (size:1440) (ours)	70.52	78.25	78.70

(b) **Results on BOVText.**

Method	MOTA (↑)	MOTP (↑)	IDF1 (↑)
EAST + CRNN [10]	-79.3	76.3	6.8
PSENet + CRNN [10]	-17.0	79.2	31.3
DB + CRNN [10]	-13.2	81.3	38.8
TransVTSpotter [10]	-1.4	82.0	43.6
CoText [11]	11.4	80.3	48.3
GoMatching (ours)	**52.9**	**87.2**	**62.6**

(c) **Results on DSText.**

Method	M-ME	MOTA (↑)	MOTP (↑)	IDF1 (↑)
TransDETR+HRNet†	Y	-28.58	80.36	26.20
SCUT-MMOCR-KS†	Y	-27.47	76.59	43.61
TextTrack†	Y	-25.09	74.95	26.38
abcmot†	Y	5.54	74.61	24.25
DA†	Y	10.51	78.97	53.45
TencentOCR†	Y	22.44	**80.82**	**56.45**
TransDETR [12]*	N	-22.63	79.73	26.43
GoMatching (ours)	N	**22.83**	80.43	46.09

(d) **Results on ArTVideo.**

Method	MOTA (↑)	MOTP (↑)	IDF1 (↑)
ArTVideo Tracking			
TransDETR [12]	54.2	67.9	70.4
GoMatching (ours)	**67.2**	**81.3**	**75.8**
ArTVideo End-to-End Spotting			
TransDETR [12]	2.8	69.7	49.3
GoMatching (ours)	**68.8**	**82.9**	**78.5**
ArTVideo-Curved Tracking			
TransDETR [12]	4.4	60.5	50.2
GoMatching (ours)	**59.5**	**76.3**	**73.5**
ArTVideo-Curved End-to-End Spotting			
TransDETR [12]	-66.7	61.9	26.9
GoMatching (ours)	**56.8**	**78.0**	**73.9**

methods have unsatisfactory recognition capabilities and limited adaptability to complex scenarios. Furthermore, as shown in Fig. 1(b), GoMatching achieves excellent performance while significantly reducing the training budget.

Some visual results are provided in Fig. 4. It shows that GoMatching performs well on straight and curved text, and even more complex scene text. More visual results (including some failure cases) and analysis are provided in the Appendix G.

4.4 Ablation Studies

We first conduct comprehensive ablation studies on ICDAR15-video to verify the effectiveness of each component. The experimental results are shown in Table 2. The impact of frame length on long-term association during inference is then studied, and the results are shown in Appendix C.

Effectiveness of Utilizing Queries. Comparing the first two rows in Table 2, we can find that using queries from the decoder of image text spotter is more beneficial for tracking than RoI features. By leveraging the unified queries from frozen DeepSolo, 0.98% and 1.05% improvements on MOTA and IDF1 are achieved. This is because queries integrate more text instance information, *i.e.*, unifying multi-scale features, text semantics, and position information, which has been proven effective in DeepSolo. Although position information is essential for tracking, it is ignored in RoI features.

Effectiveness of Rescoring Mechanism. To verify the effectiveness of the rescoring mechanism, we test three different scoring mechanisms: the original score from DeepSolo, the score recomputed by the rescoring head, and the fusion score from the rescoring mechanism. As shown in row 2 and row 3 of Table 2, the rescoring head can alleviate the performance degradation caused by the domain gap between ICDAR15-image and ICDAR15-video, and achieve gains of 1.25% and 0.97% on MOTA and IDF1, respectively. Moreover, as shown in row 4, we can observe that combining the knowledge of rescoring head learned from the new dataset with the prior knowledge of DeepSolo can further improve MOTA and IDF1 by 0.33% and 0.32%, respectively. Appendix F contains more results.

Effectiveness of LST-Matcher. In this part, we conduct three experiments to prove the effectiveness of the LST-Matcher. As shown in row 4 of Table 2, we only use LT-Matcher to associate high-score text instances in the current frame with trajectories in the tracking memory bank. In row 5, we only use ST-Matcher to associate high-score text instances in the current frame with trajectories of the previous frame. In addition, as shown in row 6, we employ both LT-Matcher and ST-Matcher to test LST-Matcher. We can easily observe that compared to LT-Matcher, LST-Matcher improves MOTA

Figure 4: **Visual results of video text spotting.** Images from top to bottom are the results on ICDAR15-video, BOVText, DSText, and ArTVideo, respectively. Text instances belonging to the same trajectory are assigned the same color.

Table 2: Impact of difference components in the proposed GoMatching. 'Query' indicates that LST-Matcher employs the queries of high-score text instances for association, otherwise RoI features. Column 'Scoring' indicates the employed scoring mechanism, in which 'O' means using the original scores from DeepSolo, 'R' means using the scores recomputed by the rescoring head, and 'F' means using the fusion scores obtained from the rescoring mechanism.

Index	Query	Scoring	LT-Matcher	ST-Matcher	MOTA (↑)	MOTP (↑)	IDF1 (↑)
1		O	✓		66.20	78.52	75.07
2	✓	O	✓		67.22	78.54	76.12
3	✓	R	✓		68.47	78.29	77.09
4	✓	F	✓		68.80	78.24	77.41
5	✓	F		✓	69.40	**78.34**	73.60
6	✓	F	✓	✓	**70.52**	78.25	**78.70**

and IDF1 by 1.72% and 1.29% respectively, while compared to ST-Matcher, the improvement on MOTA and IDF1 are 1.12% and 5.1%, respectively. In Fig. 5, we also demonstrate that using LST-Matcher can effectively mitigate the issue of ID switches caused by the strong appearance changes due to motion blur. These results validate that combining short-term and long-term information leads to more robust tracking outcomes, thereby enhancing the performance of video text spotting.

Different training strategies. To investigate the impacts of different training strategies on GoMatching, we establish three distinct settings on the ICDAR15-video dataset, as shown in Table 3. 1) We only fine-tune the tracker while keeping the image text spotter frozen (the first row of Table 3). 2) We first fine-tune the image text spotter on images extracted from ICDAR15-video and then further fine-tune the tracker with the image text spotter fixed (the second row of Table 3). 3) We jointly train the spotter's decoder and tracker of GoMatching while trying different learning rates for the decoder (the last three rows of Table 3). As shown in the first two rows of Table 3, fine-tuning the image spotter on the downstream video dataset results in a performance decline compared to the default setting. This is due to two data-related factors: **1)** minor variations in text content between frames in the same video lead to insufficient data diversity, causing the image text spotter to overfit more easily; and **2)** image blurring from camera motion reduces the quality of data available for training the image text spotter. When the image text spotter and tracker are trained simultaneously (the last three rows), the model's performance significantly decreases. Even with the decoder's learning rate close to zero (*i.e., 0.001*), there is still a 1.13% drop in IDF1. As the decoder's learning rate increases, the performance decline becomes more pronounced. This indicates that naive joint optimization of text spotting and tracking is challenging, likely due to conflicts between the two tasks. In future work,

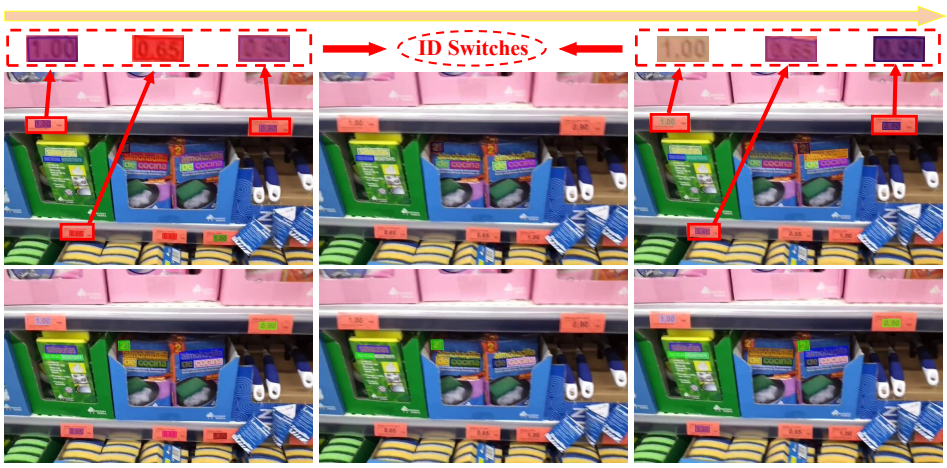


Figure 5: **Visualization results of ST-Matcher and LST-Matcher.** The first row shows the failure case that suffers from the ID switches issue when using only ST-Matcher, caused by missed detection and erroneous matching. The second row shows that LST-Matcher effectively mitigates this issue via both long and short term matching. Text instances in the same color represent the same IDs.

Table 3: Results of GoMatching under various training settings on the ICDAR15-video dataset. 'Only Image Spotter' and 'Only Tracker' refer to fine-tuning either the image spotter or tracker with another module fixed. 'End-to-End' denotes that training the image spotter and tracker in an end-to-end manner. '0.001', '0.01' and '0.1' correspond to the ratios of the learning rate employed by the decoder of the image text spotter relative to the base learning rate. Due to constraints in training resources, we only optimize the parameters of the decoder component for the image text spotter.

Index	Training Setting	MOTA (↑)	MOTP (↑)	IDF1 (↑)
1	'Only Tracker'	**72.04**	78.53	**80.11**
2	First 'Only Image Spotter', Then 'Only Tracker'	70.82	78.09	79.64
3	'End-to-End', Image Spotter's Decoder ('0.001')	71.48	**79.14**	78.98
4	'End-to-End', Image Spotter's Decoder ('0.01')	70.15	78.17	77.67
5	'End-to-End', Image Spotter's Decoder ('0.1')	68.03	75.46	77.16

it is worth trying to establish larger, more diverse video text spotting datasets and to explore more effective multi-task optimization strategies.

5 Conclusion

In this paper, we propose a simple yet strong baseline, termed GoMatching, for video text spotting. GoMatching harnesses the talent of an off-the-shelf query-based image text spotter and only needs to tune a lightweight tracker, effectively addressing the limitations of previous SOTA methods in recognition. Specifically, we design the rescoring mechanism and LST-Matcher to adapt the image text spotter to unseen video datasets while empowering GoMatching with excellent tracking capability. Moreover, we establish a novel test set ArTVideo for the evaluation of video text spotting models on arbitrary-shaped text, filling the gap in this area. Experiments on public benchmarks and ArTVideo demonstrate the superiority of our GoMatching in terms of both spotting accuracy and training cost.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFC2705700, in part by the National Natural Science Foundation of China under Grant U23B2048, 62076186 and 62225113, in part by the Innovative Research Group Project of Hubei Province under Grant 2024AFA017, and in part by the Science and Technology Major Project of Hubei Province under Grant 2024BAB046. Dr Tao's research is partially supported by NTU RSR and Start Up Grants. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

References

[1] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4065–4080, 2021.

[2] Chongsheng Zhang, Yuefeng Tao, Kai Du, Weiping Ding, Bin Wang, Ji Liu, and Wei Wang. Character-level street view text spotting based on deep multisegmentation network for smarter autonomous driving. *IEEE Transactions on Artificial Intelligence*, 3(2):297–308, 2021.

[3] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. Spts v2: single-point scene text spotting. *arXiv preprint arXiv:2301.01635*, 2023.

[4] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9519–9528, 2022.

[5] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023.

[6] Mingxin Huang, Jiaxin Zhang, Dezhi Peng, Hao Lu, Can Huang, Yuliang Liu, Xiang Bai, and Lianwen Jin. Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19495–19505, 2023.

[7] Xiaobing Wang, Yingying Jiang, Shuli Yang, Xiangyu Zhu, Wei Li, Pei Fu, Hua Wang, and Zhenbo Luo. End-to-end scene text recognition in videos based on multi frame tracking. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1255–1260. IEEE, 2017.

[8] Zhanzhan Cheng, Jing Lu, Yi Niu, Shiliang Pu, Fei Wu, and Shuigeng Zhou. You only recognize once: Towards fast video text spotting. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 855–863, 2019.

[9] Zhanzhan Cheng, Jing Lu, Baorui Zou, Liang Qiao, Yunlu Xu, Shiliang Pu, Yi Niu, Fei Wu, and Shuigeng Zhou. Free: A fast and robust end-to-end video text spotter. *IEEE Transactions on Image Processing*, 30:822–837, 2020.

[10] Weijia Wu, Yuanqiang Cai, Debing Zhang, Sibo Wang, Zhuang Li, Jiahong Li, Yejun Tang, and Hong Zhou. A bilingual, openworld video text dataset and end-to-end video text spotter with transformer. *arXiv preprint arXiv:2112.04888*, 2021.

[11] Wejia Wu, Zhuang Li, Jiahong Li, Chunhua Shen, Hong Zhou, Size Li, Zhongyuan Wang, and Ping Luo. Real-time end-to-end video text spotter with contrastive representation learning. *arXiv preprint arXiv:2207.08417*, 2022.

[12] Weijia Wu, Yuanqiang Cai, Chunhua Shen, Debing Zhang, Ying Fu, Hong Zhou, and Ping Luo. End-to-end video text spotting with transformer. *International Journal of Computer Vision*, 132 (9):4019–4035, 2024.

[13] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[14] 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*, pages 213–229. Springer, 2020.

[15] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.

[16] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22056–22065, 2023.

[17] En Yu, Tiancai Wang, Zhuoling Li, Yuang Zhang, Xiangyu Zhang, and Wenbing Tao. Motrv3: Release-fetch supervision for end-to-end multi-object tracking. *arXiv preprint arXiv:2305.14298*, 2023.

[18] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022.

[19] Weijia Wu, Yuzhong Zhao, Zhuang Li, Jiahong Li, Mike Zheng Shou, Umapada Pal, Dimosthenis Karatzas, and Xiang Bai. Icdar 2023 video text reading competition for dense and small text. *arXiv preprint arXiv:2304.04376*, 2023.

[20] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer, 2020.

[21] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

[22] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022.

[23] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8771–8780, 2022.

[24] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020.

[25] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Zhibo Yang, Tong Lu, and Chunhua Shen. Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5349–5367, 2021.

[26] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8048–8064, 2021.

[27] Yair Kittenplon, Inbal Lavi, Sharon Fogel, Yarin Bar, R Manmatha, and Pietro Perona. Towards weakly-supervised text spotting using a multi-task transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4613, 2022.

[28] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo++: Let transformer decoder with explicit points solo for multilingual text spotting. *arXiv preprint arXiv:2305.19957*, 2023.

[29] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[31] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[32] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016.

[33] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[35] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11474–11481, 2020.

[36] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, pages 1483–1498, 2019.

[37] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In *European Conference on Computer Vision*, pages 178–196. Springer, 2022.

[38] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[39] Yuzhe Gao, Xing Li, Jiajian Zhang, Yu Zhou, Dian Jin, Jing Wang, Shenggao Zhu, and Xiang Bai. Video text tracking with a spatio-temporal complementary model. *IEEE Transactions on Image Processing*, pages 9321–9331, 2021.

[40] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90: 337–345, 2019.

[41] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013.

A More Details of ArTVideo

Due to the scarcity of curved text instances within existing video text spotting datasets, it is infeasible to evaluate the performance of video text spotting models on curved text. To fill this gap, we collected a test set named ArTVideo containing 20 video clips with a total of 884 frames, in which 18 videos were collected from YouTube and 2 videos from the BOVText test set. ArTVideo contains 6,526 text instances, including 4,632 straight text instances and 1,894 curved text instances, *i.e.*, curved text accounts for about 30%. As shown in Fig. 6, we provide high-quality word-level annotations for both straight and curved text in two different annotated ways. The straight text is labeled with quadrilaterals, while for curved text, we follow the CTW1500 [40] and adopt a polygon with 14 points to annotate the text contour. More statistics about ArTVideo are shown in Fig. 7.

Figure 6: **Visual examples from our ArTVideo**. The straight and curved text are labeled with quadrilaterals and polygons, respectively. The same background color in different frames (columns) denotes the same instance.

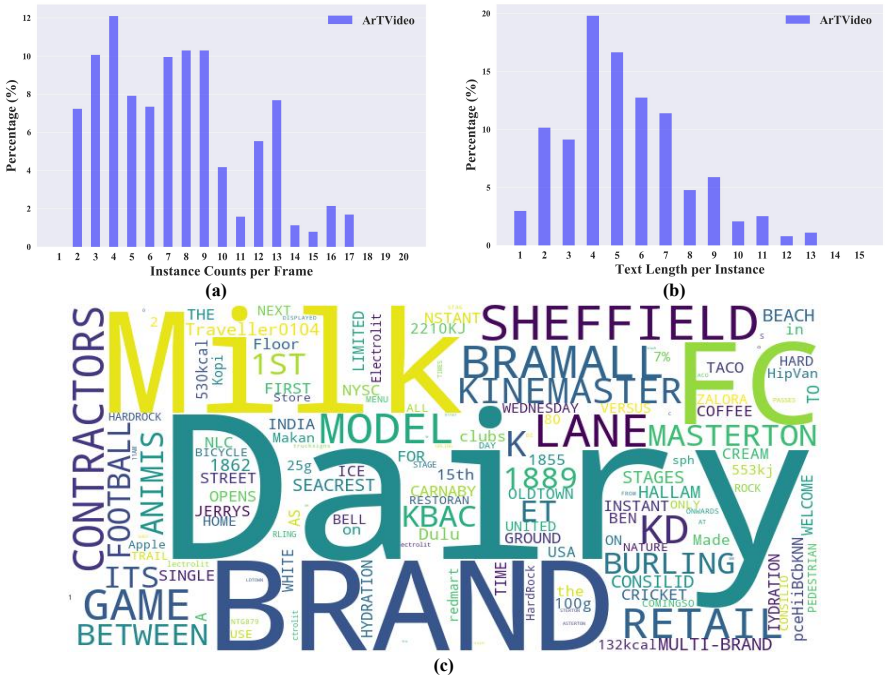


Figure 7: **Statistics of ArTVideo.** (a) and (b) show the distribution of text instance numbers in each frame and the distribution of the text length of each instance, respectively. (c) shows the word cloud of text annotations in ArTVideo.

Table 4: Ablation studies on the number of frames (T) for long-term association in LT-Matcher, and the max number of history frames in tracking memory bank is $H = T - 1$). Experiments are conducted on ICDAR15-video and the best results are marked in **bold**.

Number T	MOTA (↑)	MOTP (↑)	IDF1 (↑)
$T = 32$	70.13	78.07	78.24
$T = 16$	70.33	78.25	78.60
$T = 8$	70.44	78.25	**78.70**
$T = 6$	**70.52**	78.25	**78.70**
$T = 4$	70.51	**78.27**	78.16

Table 5: **Results of different score fusion strategies on ICDAR5-video.** 'Mean', 'Geo-mean', and 'Maximum' denote the arithmetic mean, geometric mean, and the maximum score fusion strategies, respectively. The best results are highlighted in **bold**.

Strategy	MOTA (↑)	MOTP (↑)	IDF1 (↑)
Mean	70.46	78.38	78.29
Geo-mean	70.29	**78.39**	78.26
Maximum	**70.52**	78.25	**78.70**

Table 6: **Results of using different image sizes on ICDAR15-video.** 'Size' means the size of the shorter side of the input image during inference. The best results are highlighted in **bold**.

Method	MOTA (↑)	MOTP (↑)	IDF1 (↑)	FPS (↑)
TransDETR(Size: 800)	60.96	74.61	72.80	12.69
GoMatching(Size: 800)	68.51	77.52	76.59	**14.41**
GoMatching(Size: 1000)	**72.04**	**78.53**	**80.11**	10.60

Table 7: **Comparison between TransDETR and GoMatching.** 'T-Para.' and 'A-Para.' denote the number of all parameters and the trainable parameters in each model, respectively.

Method	#T-Para. (M)	#A-Para. (M)
TransDETR	39.35	39.58
GoMatching	32.79	75.38

B More Details of Inference Settings

Since there is no training set in ArTVideo, we directly use the model trained on ICDAR15-video to evaluate the generalization ability of GoMatching to arbitrary-shaped text. The association score threshold is set to 0.2. For ICDAR15-video, we set the shorter size of the input image to 800, 1000, and 1440, with 800 aligned with the setting in TransDETR and 1440 aligned with the setting in DeepSolo. As for BOVText, DSText, and ArTVideo, the shorter sizes are set to 1000, 1280, and 1440, respectively. All the ablation studies are conducted on the setting of 1440.

C Impact of the Frame Number in LT-Matcher

In Table 4, we further study and analyze the impact of the number of frames for long-term association in LT-Matcher during inference. For the text spotting task, since a single frame may have a large number of text instances, sometimes reaching hundreds, excessive historical frame information would weaken the discrimination of text instance features, resulting in erroneous matching results. Therefore, we conduct a hyper-parameter search and find that the optimal frame number is 6.

D Comparison of Different Score Fusion Strategies

To investigate the impact of the score fusion strategy in the rescoring mechanism, we further evaluate two other strategies: (1) the arithmetic mean score fusion strategy and (2) the geometric mean score fusion strategy, denoted as mean and geo-mean, respectively. The arithmetic mean score fusion strategy takes the average of the scores from the image text spotter and the rescoring head as the final score for each query, while the geo-mean score fusion strategy uses the geometric mean. These two strategies can be formulated as:

$$c_{mean} = (c_o + c_r)/2, \tag{13}$$

$$c_{geo-mean} = \sqrt{c_o * c_r}, \tag{14}$$

where c_{mean} and $c_{geo-mean}$ denote the final score of the two strategies, respectively. c_o and c_r are the scores from the image text spotter and the rescoring head, respectively.

From Table 5, we can see that employing the maximum score fusion strategy achieves the best performance on MOTA and IDF1 among all the strategies. As for the other two strategies, extremely low confidence scores from the image text spotter may result in low final scores, probably leading to missed detections. Therefore, we adopt the maximum score fusion strategy in the rescoring mechanism of GoMatching by default.

Figure 8: **Visual comparison of GoMatching with and without the rescoring mechanism.** The values in parentheses indicate the confidence scores of the detected text. '$\rightarrow$' points to the filtered texts due to low confidence without using rescoring. '$\rightarrow$' points to the text whose confidence score has been improved by the rescoring mechanism. The rescoring mechanism increases the confidence scores of small texts and blurry texts, preventing them from being filtered out by the threshold and thereby cultivating a better tracking candidate pool.

Table 8: **Video text detection performance on ICDAR2013-video [41].** The best results are highlighted in **bold**.

Method	Precision (↑)	Recall (↑)	F-measure (↑)
Free	79.7	68.4	73.6
TransDETR	80.6	70.2	75.0
GoMatching w/o rescoring	**92.4**	65.7	76.8
GoMatching	89.5	**74.8**	**81.5**

Table 9: **Comparison results of detection AP on the ICDAR13-video between with and without the rescoring mechanism.**

Method	AP	AP_S	AP_M	AP_L
w/o rescoring	26.2	11.6	40.1	49.8
w/ rescoring	29.3 (+3.1)	15.5 (+3.9)	42.7 (+2.6)	51.9 (+2.1)

E More Comparisons between TransDETR and GoMatching

In Table 6, we provide the results of using three different image sizes in GoMatching during inference on ICDAR15-video. In the first row of Table 6, the shorter side of the input image is set to 800, which is the same as the default setting of TransDETR. It is evident from Table 6 that GoMatching significantly outperforms TransDETR in all settings. Meanwhile, it also shows that GoMatching outperforms TransDETR under its default setting in terms of both inference speed and spotting

Figure 9: **More visualization results of TransDETR and GoMatching.** Failure cases of text detection and recognition are highlighted with ellipses and rectangles, respectively.

accuracy. With the increase in image size (*e.g.*, 1000), GoMatching offers better spotting performance at the cost of decreased inference speed.

Moreover, we compare the number of all parameters and the trainable parameters of GoMatching and TransDETR, as shown in Table 7. It is noteworthy that GoMatching requires a much smaller training budget than TransDETR owing to its simpler architecture design, making it a simple but strong baseline for video text spotting.

F More Results for the Rescoring Mechanism

In Table 8, we provide the video text detection results of GoMatching on ICDAR13-video [41] and compare them with Free and TransDETR. As shown in the table, without the rescoring mechanism, GoMatching relies on the original results from DeepSolo, resulting in a 4.5% decrease on Recall and only a 1.8% improvement on F-measure compare to TransDETR. This is due to the domain gap between image data and video data, directly using an image spotter leads to a low confidence and consequently low Recall on video data. When encompasses the rescoring mechanism, GoMatching achieves a 9.1 improvement on Recall compared to DeepSolo and a 6.5% F-measure enhancement compared to TransDETR. These improvements highlight the effectiveness of rescoring mechanism in alleviating the domain gap and leading to a better tracking candidate pool. The impressive results of GoMatching are not merely attributed to the introduction of a robust image text spotter.

To further explore how the rescoring mechanism eases the domain gap, we calculate the AP of detection results on ICDAR13-video, as shown in the Table 9. Incorporating the rescoring mechanism effectively improves the detection performance of DeepSolo on video datasets, particularly for small text, resulting in a 3.9% improvement on AP_S. We also present more visual results to embody the potency of rescoring mechanism in Fig. 8.

G More Visualization Results

We present more visualization results of TransDETR and GoMatching in Fig. 9, including some failure cases. It can be observed that GoMatching exhibits a significant improvement in text recognition performance compared to TransDETR. It should be noted that GoMatching may also experience failures due to the image-video domain gap and extreme cases, such as very small text instances and significant motion blur. These issues can be mitigated by employing a stronger text spotter with a more representative backbone and training on larger, more diverse datasets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of GoMatching in Appendix G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed information about the training and inference settings in Section 4.2 and Appendix B, along with the release of the code and trained models, ensuring that all necessary information for reproducing the main experimental results is disclosed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided the data and code, with sufficient instructions for open access in supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We specify the training and test details in Section 4.2 and Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The experiments reported in the paper do not include error bars. Instead, we focus on providing comprehensive results and analyses to demonstrate the performance of our approach.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide relevant information in Section 4.2 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: No ethical issues involved.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Our paper aiming to improve performance and efficiency in text spotting from videos, and does not have any direct negative applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All data and models utilized in this paper are publicly available and appropriately cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We are submitting the code for GoMatching in the supplementary material and providing a "readme.md" file to reproduce our results.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.