

Sketch-based Video Object Localization

Sangmin Woo¹ So-Yeong Jeon^{1,2} Jinyoung Park¹ Minji Son³ Sumin Lee¹ Changick Kim¹
¹KAIST ²Korea Agency for Defense Development ³LG Electronics

¹{smwoo95, presentover, jinyoungpark, suminlee94, changick}@kaist.ac.kr ³minji13.son@lge.com

Abstract

We introduce *Sketch-based Video Object Localization (SVOL)*, a new task aimed at localizing spatio-temporal object boxes in video queried by the input sketch. We first outline the challenges in the SVOL task and build the *Sketch-Video Attention Network (SVANet)* with the following design principles: (i) to consider temporal information of video and bridge the domain gap between sketch and video; (ii) to accurately identify and localize multiple objects simultaneously; (iii) to handle various styles of sketches; (iv) to be classification-free. In particular, SVANet is equipped with a *Cross-modal Transformer* that models the interaction between learnable object tokens, query sketch, and video through attention operations, and learns upon a per-frame set matching strategy that enables frame-wise prediction while utilizing global video context. We evaluate SVANet on a newly curated SVOL dataset. By design, SVANet successfully learns the mapping between the query sketches and video objects, achieving state-of-the-art results on the SVOL benchmark. We further confirm the effectiveness of SVANet via extensive ablation studies and visualizations. Lastly, we demonstrate its transfer capability on unseen datasets and novel categories, suggesting its high scalability in real-world applications. Codes are available at <https://github.com/sangminwoo/SVOL>.

1. Introduction

A sketch is worth a thousand words. It can even convey ideas that are hard to explain in words. Due to the concise and abstract nature of the sketch, it can be illustrative, making it an excellent tool for a variety of applications [2, 4, 9, 13, 19, 29, 32, 34]. Meanwhile, query-based localization is one of the long-sought goals for visual understanding. The literature has been studied at a variety of query types (e.g., image, language, sketch) and domains (e.g., image, video) [1, 5, 8, 11, 14, 17, 21, 25, 27, 30, 33]. While numerous studies have shown remarkable results using image or language as query, both have their own limitations. Images containing a specific object of interest may be difficult to collect due to privacy or copyright issues [2], and

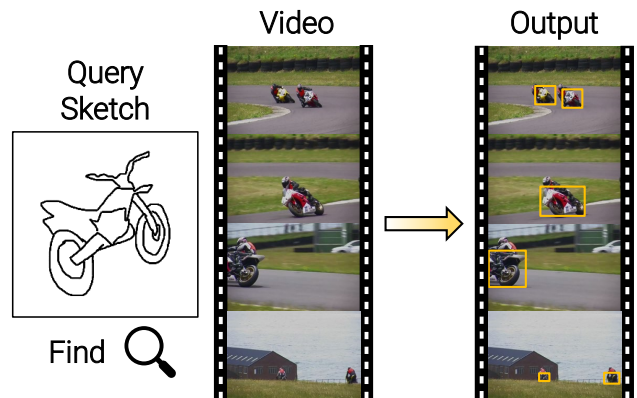


Figure 1. **Illustration of the SVOL task.** Given a query sketch, the goal is to find all *object boxes* (colored in yellow) spatio-temporally that match the sketch object in a video. Query sample is randomly drawn from *Sketchy* dataset.

the utility of language is limited as it varies per country. As an alternative, using sketch as a query brings several advantages. It allows for immense expressive flexibility and can transcend language barriers [16]. Moreover, due to the recent spread of touchscreen devices (e.g., smartphones, tablets), sketches have become easier to obtain than ever [26]. Amid the explosive growth of video data, sketch has emerged as an appealing candidate for user interface in online video platforms thanks to these properties. Despite its promise, the use of sketch as query for object localization in the video domain has not yet been explored.

In this work, we propose a new task called *Sketch-based Video Object Localization (SVOL)* that aims to localize objects in videos with the query sketch (see Fig. 1). We first identify several challenges in SVOL, including but not limited to: (i) As objects move, they can generate motion blurs or occlude parts of other objects, thus distorting their appearances [15]. Moreover, objects in the scene may suddenly disappear, or objects that were not in the scene may suddenly appear. These dynamic changes over time complicate the matching of sketch to its corresponding objects. (ii) Multiple objects can appear in a video. Therefore, it is important not only to accurately differentiate between the target objects from multiple objects belonging to different categories, but

also to find all objects that match the sketch query simultaneously. **(iii)** As for sketch, a single object can be drawn in various ways [23]. Unlike natural videos, sketches lack color, texture, and background information, resulting in a high degree of freedom. This allows sketches to be drawn in a variety of styles (*i.e.*, different abstraction levels). **(iv)** There should be no explicit category prediction (*i.e.*, no fixed classes) in the SVOL system. As with all query-based localization tasks, SVOL requires finding the best matching objects given a query sketch (not the category itself).

Driven by this analysis, we propose SVANet that serves as a strong baseline for the SVOL task. SVANet takes extracted video and sketch representations as inputs and predicts box coordinates and objectness scores end-to-end. Our SVANet is built on several design principles: **First**, we propose a novel *Cross-modal Transformer (CMT)* that not only closes the domain discrepancy between sketch and video but also models video temporal context. We equip CMT with four attention operations [28] to leverage their strong relational modeling capability. By design, CMT emphasizes important content by learning the correlation between sketch and video representations, and incorporates temporal context by modeling intra-content relationships. Also, CMT takes object tokens as inputs and transforms them into predictions of box coordinates and objectness scores by learning their internal interactions and by referring to joint sketch-video representations. **Second**, we formulate the SVOL task as a set prediction problem [3] and employ a *per-frame set matching* strategy. We predict all bounding boxes across the video frames, and find the best matching between predicted and ground truth boxes that minimizes the matching cost. The overall training loss is then defined based on the matching results. Instead of matching whole video-level results with video-level ground truths, we perform set matching frame-by-frame. This enables the prediction of multiple objects in parallel while utilizing the global video context. **Third**, SVANet is designed to be compatible with various sketch styles. SVANet learns to embed the sketch objects of the same category into a similar subspace of a high-dimensional latent space, regardless of differences in sketch styles (*e.g.*, shape, pose, line thickness, *etc.*). This style-agnostic property enables SVANet to generalize well on unseen sketch datasets with varying degrees of abstraction. **Last**, SVANet has no explicit classification in the pipeline. This allows SVANet to learn the mapping between sketch and video objects based on implicit similarity (*e.g.*, symbolic meaning, appearance, *etc.*). This classification-free property of SVANet extends its applicability to any kind of free-form sketches, allowing us to query over novel object classes.

To benchmark our approach and show the potential of using sketch as query, we present a new SVOL dataset curated from the video dataset, ImageNet-VID [22], and three sketch datasets with varying degrees of abstraction (see Fig. 2):







	Sketchy	TU-Berlin	QuickDraw
cat			
dog			
Abstract	Low	Medium	High

Figure 2. **Sketch datasets comparison.** **Sketchy** [24] is the most realistic since it is drawn after photographic objects, **QuickDraw** [10] has the highest level of abstraction due to limited drawing time (< 20 secs), and **TU-Berlin** [6] lies halfway between them.

Sketchy [24], **TU-Berlin** [6], and **QuickDraw** [10]. We show that SVANet outperforms the strong image-level baseline (Sketch-DETR) [21] by a significant margin: $\sim 29.4\%$, 17.7% , and 16.8% improvement of mIoU using **Sketchy**, **TU-Berlin**, and **QuickDraw** sketch datasets, respectively. This implies that SVANet effectively resolves the limitation of image-level baselines with temporal video context. Moreover, we verify the effectiveness of several design choices of SVANet through extensive ablation studies and analyze its behavior with several visualizations. Finally, we evaluate transfer performances of SVANet on unseen datasets (with different abstraction levels or sketch styles) and novel categories that are unseen during training. The results demonstrate that SVANet is robust to style variations and that the learned sketch-video mapping function generalizes well to novel classes of sketches. These appealing properties are ideal for several query-based applications in practice, such as large-scale video platforms, in that the system can flexibly respond to diverse inputs from users.

2. Method

We begin by describing the SVOL task and present an end-to-end trainable SVANet that predicts a set of objects based on dense pair-wise relation modeling. Next, we introduce a per-frame set matching strategy that imposes a unique match between predicted and ground truth boxes at each frame; then, define an overall training loss. An overview of SVANet is depicted in Fig. 3.

SVOL task definition. Given a query sketch \mathcal{S} and a video \mathcal{V} , the goal of SVOL is to find all spatio-temporal boxes \mathcal{Y} that *match* the sketch object in the video. We consider the video as a sequence of L frames, $\mathcal{V} = [V_i]_{i=1}^L$, and aim to find all boxes $\mathcal{Y} = [B_i]_{i=1}^L$ over the video frames, where $B_i \in \mathbb{R}^{K_i \times 4}$ is a set of bounding boxes at video frame V_i , $B_i = \{b_i^j\}_{j=1}^{K_i}$. The number of boxes K_i at frame V_i can vary throughout the video, since objects can be occluded, disappear or appear in the scene. We predict a total of N bounding boxes across L frames, $\hat{\mathcal{B}} = [\hat{B}_i]_{i=1}^L$, M boxes per frame, $\hat{B}_i = \{\hat{b}_i^j\}_{j=1}^M$, where $N = L \times M$. The predictions are considered as *correct* if IoU between the predicted box \hat{b}

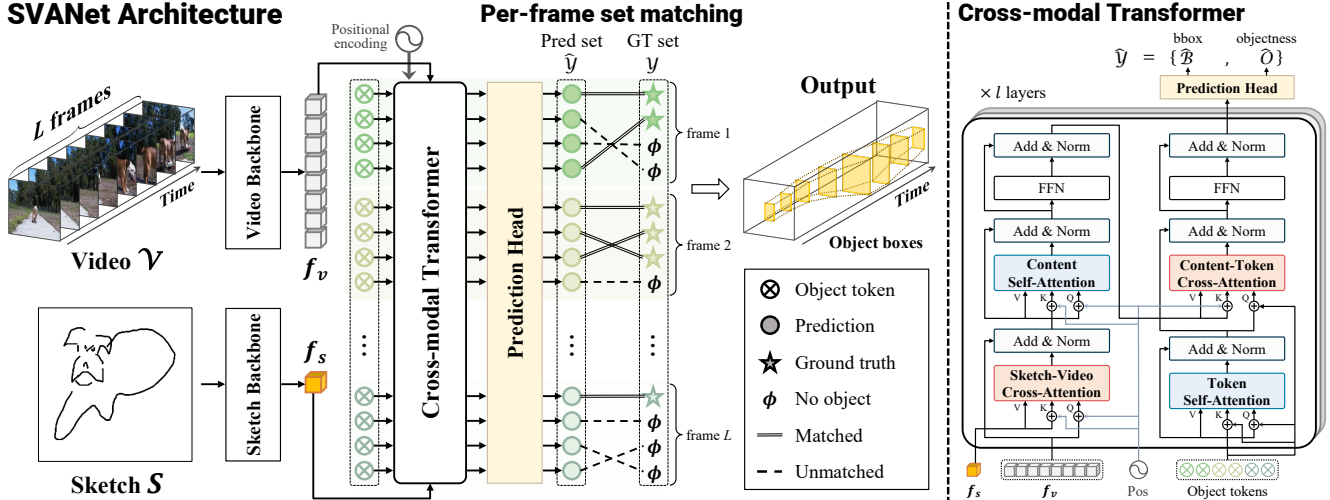


Figure 3. **Overview of SVANet.** Given a video \mathcal{V} and a query sketch \mathcal{S} , SVANet processes them in a separate encoding pipeline, yielding a sequence of frame representations f_v and a sketch representation f_s . The Cross-modal Transformer (CMT) then takes f_v , f_s , and a set of learnable object tokens as inputs. Through the CMT layers, object tokens learn interactions between themselves and attend to sketch-video joint representation to produce accurate predictions. **Per-frame set matching:** During training, SVANet finds the best matching that minimizes the matching cost (see Eq. (9)) between the prediction set and the ground truth set for each frame. To assign a unique matching between the two sets, the ground truth set is padded with additional No object (ϕ) elements. The overall loss is defined based on the matching results (see Eq. (11)). As a result, SVANet outputs the spatio-temporal object boxes. **Cross-modal Transformer:** In CMT, sequences of representations are added with positional encoding before every attention operation. CMT first highlights important contents by learning the correspondence between sketch f_s and video f_v representations, and models intra-content relationships. CMT then transforms the object token set into a set of predictions (box coordinates and objectness scores) by learning token-token interactions and referring to joint sketch-video representations. More details are in Sec. 2. Best viewed in color.

and the ground truth box b is higher than the threshold μ . The bounding box is defined as a 4D vector normalized w.r.t. the frame resolution: $b \in [0, 1]^4$. We also predict the likelihoods that the predicted boxes contain the target object, referred to as objectness scores \hat{O} , where each element $\hat{o} \in [0, 1]$. In short, the predictions are a set of bounding boxes and their corresponding objectness scores: $\hat{\mathcal{Y}} = \{\hat{\mathcal{B}}, \hat{\mathcal{O}}\}$. As we view SVOL as a set prediction problem, we find the best matching between the ground truth set \mathcal{Y} and the prediction set $\hat{\mathcal{Y}}$.

As we set the SVOL problem as category-level localization, the system is trained to perform the consistent bounding box localization for sketches belonging to the same category, regardless of variations in shape or pose. This allows the system to operate robustly, even in the presence of different levels of abstraction or diverse styles in the sketches. However, it is worth noting that there is *no* explicit category prediction inside the system, instead it relies on implicit similarity (e.g., symbolic meaning, appearance, etc.) to learn sketch-video object matching.

2.1. SVANet Architecture

SVANet is designed to address the challenge of bridging the gap between two distinct modalities, sketch and natural video, in order to perform object localization. The system incorporates attention operations that consider a wide range of contexts and inter-dependencies between elements within the input sequences. This leads our system to acquire the

capability to learn powerful representations of the input sequences and delivers accurate video object localization using sketches as queries.

Video & sketch backbones. A video, represented by a sequence of frames, $\mathcal{V} \in \mathbb{R}^{L \times C_0 \times H_0 \times W_0}$, where $L = 32$, $C_0 = 3$, $H_0 = W_0 = 224$, is initially processed using the ResNet-50 architecture [7], generating high-dimensional feature maps $f_v \in \mathbb{R}^{L \times C \times H \times W}$, where $C = 512$, $H = W = 7$. Likewise, a sketch \mathcal{S} is processed using ResNet-18 [7], followed by a spatial pooling operation that compresses it into 1D representation $f_s \in \mathbb{R}^C$. Finally, the outputs f_v and f_s are passed through the Cross-modal Transformer. To address the sparse nature of sketch information, we use a lighter CNN backbone (ResNet-18) compared to the video (ResNet-50).

Cross-modal Transformer & prediction head. In addition to f_v and f_s , the Cross-modal Transformer (CMT) takes a set of N learnable embeddings initialized with random weights, which we refer to as *object tokens*, and transforms them into a set of N predictions.

CMT consists of l layers, and each layer contains four attention operations: (i) *Sketch-Video Cross-Attention (SVCA)* assigns higher attention weights to the important elements of the input sequence (video patches), that are relevant for accurate bounding box localization based on the input sketch query. This is achieved by modeling the inter-modality relationship between video f_v and sketch f_s representations.

SVCA bridges the gap between sketches and videos by effectively integrating the information from both modalities. (ii) *Content Self-Attention (CSA)* is responsible for modeling the temporal relationship between the elements in the input sequence (*i.e.*, output of SVCA). By considering the pairwise relationship of these elements, CSA enables a more comprehensive understanding of the broader video context. (iii) *Token Self-Attention (TSA)* receives object tokens as input and models interactions between them, enabling them to globally reason about all objects. (iv) *Content-Token Cross-Attention (CTCA)* transforms object tokens to meaningful outputs by relating them with contextual representation of content (*i.e.*, output of CSA). Since the attention operations are permutation-invariant (*i.e.*, produce the same output regardless of the order of elements in the input sequence), we supplement the input sequence with temporal order information by adding absolute positional encoding prior to every attention operation (except for TSA; we instead add object tokens to the input sequence of each TSA operation).

All attention operations in CMT are in the form of Multi-Head Attention with 8 heads (*i.e.*, $k = 8$). Let a video representation $f_v \in \mathbb{R}^{L \times C \times H \times W}$ as $\mathbf{v}^{(0)}$ and a sketch representation $f_s \in \mathbb{R}^{1 \times C}$ as \mathbf{s} . Given $\mathbf{v}^{(0)}$ and \mathbf{s} the i -th CMT layer calculates:

$$\mathbf{x}^{(i)} = \text{LN}(\text{SVCA}^{(i)}(\mathbf{v}^{(i)}, \mathbf{s}, \mathbf{s}) + \mathbf{v}^{(i)}), \quad (1)$$

$$\mathbf{y}^{(i)} = \text{LN}(\text{CSA}^{(i)}(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}, \mathbf{x}^{(i)}) + \mathbf{x}^{(i)}), \quad (2)$$

$$\mathbf{v}^{(i+1)} = \text{LN}(\text{FFN}_1^{(i)}(\mathbf{y}^{(i)}) + \mathbf{y}^{(i)}), \quad (3)$$

$$\mathbf{p}^{(i)} = \text{LN}(\text{TSA}^{(i)}(\mathbf{r}^{(i)}, \mathbf{r}^{(i)}, \mathbf{r}^{(i)}) + \mathbf{r}^{(i)}), \quad (4)$$

$$\mathbf{q}^{(i)} = \text{LN}(\text{CTCA}^{(i)}(\mathbf{p}^{(i)}, \mathbf{v}^{(i+1)}, \mathbf{v}^{(i+1)}) + \mathbf{p}^{(i)}), \quad (5)$$

$$\mathbf{r}^{(i+1)} = \text{LN}(\text{FFN}_2^{(i)}(\mathbf{q}^{(i)}) + \mathbf{q}^{(i)}), \quad (6)$$

where LN is layer normalization and FFN is 2-layer feed-forward network. Here, $\mathbf{r}^{(0)} = \mathbf{0}_{N \times C}$ ($N \times C$ -sized zero matrix), thus TSA operation Eq. (4) can be omitted in the first CMT layer.

TSA (Eq. (4)) and CTCA operations (Eq. (5)) slightly differ with the standard QKV attention in that they consider the object tokens tkn as learnable positional encoding for the query (\mathbf{q}) inputs, *i.e.*, \mathbf{q} is added with tkn instead of fixed positional encoding.

$$\mathbf{Q} = (\mathbf{q} + \text{tkn})\mathbf{W}_{\mathbf{q}}. \quad (7)$$

In addition, since TSA is Self-Attention operation ($\mathbf{q} = \mathbf{k} = \mathbf{v} = \mathbf{r}^{(i)}$), tkn is also used as positional encoding for the key

(\mathbf{k}) input in TSA.

$$\mathbf{K} = (\mathbf{k} + \text{tkn})\mathbf{W}_{\mathbf{k}}. \quad (8)$$

The subsequent processes are the same as standard QKV attention.

We go through l CMT layers, and the final CMT output $r^{(l)}$ is fed into two separate linear layers (*i.e.*, prediction heads) to obtain a set of bounding box coordinates $\hat{\mathcal{B}}$ and objectness scores $\hat{\mathcal{O}}$, respectively.

2.2. SVOL as a Set Prediction

In this work, we formulate SVOL as a set prediction problem. In practice, we adopt a Hungarian algorithm [12] to find an optimal matching between predictions and ground truths in a way that minimizes the matching cost. The overall loss function is defined based on the matching results.

Per-frame set matching. SVANet transforms N object tokens to N predictions (bounding boxes and objectness scores). Here, we make each of the N/L (hereafter M) tokens to be responsible for predicting the results of each frame V_i by performing per-frame set matching. This allows SVANet to predict results per frame while being able to access global context information across the video. We formally describe the process in the following.

A set of ground truth bounding boxes \mathcal{Y} can be seen as a sequence of L subsets, where i -th subset has K_i elements: $[\{b_i^j\}_{j=1}^{K_i}]_{i=1}^L$. Likewise, we evenly divide a prediction set $\hat{\mathcal{Y}} = \{\hat{\mathcal{B}}, \hat{\mathcal{O}}\}$ of size N into L subsets having M elements each: $[\{\hat{y}_i^j\}_{j=1}^M]_{i=1}^L$, where $\hat{y}_i^j = (\hat{b}_i^j, \hat{o}_i^j)$. Hereafter, we denote the i -th subset of ground truths as Y_i and that of predictions as \hat{Y}_i for conciseness. The size of \hat{Y}_i is assumed to be larger than the size of Y_i : $M > K_i$. Since the Hungarian algorithm pairs the elements of two sets one by one, we pad Y_i with No object (\emptyset) to match the size of M . For every single i (from $i = 1$ to $i = L$), we seek for the best one-to-one matching between Y_i and \hat{Y}_i using a Hungarian algorithm. Formally, in the i -th prediction subset, let $\hat{y}_i^{\sigma_i(j)}$ be the j -th element under a permutation of M elements $\sigma_i \in \mathfrak{S}_i(M)$. We now define the pair-wise matching cost \mathcal{C} as:

$$\mathcal{C}(b_i^j, \hat{y}_i^{\sigma_i(j)}) = -\mathbb{1}_{\{b_i^j \neq \emptyset\}} \hat{o}_i^{\sigma_i(j)} + \mathbb{1}_{\{b_i^j \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i^j, \hat{b}_i^{\sigma_i(j)}). \quad (9)$$

Note that the No object paddings in the ground truth are not considered when calculating the matching cost. For every i , we aim to find the optimal assignment $\sigma_i^* \in \mathfrak{S}_i(M)$ that pairs the predictions and ground truths at the lowest cost:

$$\sigma_i^* = \underset{\sigma_i \in \mathfrak{S}_i(M)}{\text{argmin}} \sum_{j=1}^M \mathcal{C}(b_i^j, \hat{y}_i^{\sigma_i(j)}). \quad (10)$$

Overall loss. Based on the matching results, our set prediction loss $\mathcal{L}_{set}(\mathcal{Y}, \hat{\mathcal{Y}})$ is defined as:

$$\sum_{i=1}^L \sum_{j=1}^M \left[-\lambda_{\text{obj}} \log \hat{\sigma}_i^{\sigma^*(j)} + \mathbb{1}_{\{b_i^j \neq \emptyset\}} \mathcal{L}_{\text{box}}(b_i^j, \hat{b}_i^{\sigma^*(j)}) \right], \quad (11)$$

where $\lambda_{\text{obj}} \in \mathbb{R}$ is a loss coefficient for objectness scores. Here, the log-probability of the No object paddings (\emptyset) is scaled down by a factor of 10 to strike a balance between object and no-object.

The box loss \mathcal{L}_{box} is defined as a linear combination of ℓ_1 loss and the generalized IoU (gIoU) loss [20]:

$$\mathcal{L}_{\text{box}}(b_i^j, \hat{b}_i^{\sigma^*(j)}) = \lambda_{\ell_1} \mathcal{L}_{\ell_1}(b_i^j, \hat{b}_i^{\sigma^*(j)}) + \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i^j, \hat{b}_i^{\sigma^*(j)}), \quad (12)$$

where $\lambda_{\ell_1} \in \mathbb{R}$ and $\lambda_{\text{iou}} \in \mathbb{R}$ are balancing hyperparameters. While both losses have the same goal, object localization, the ℓ_1 loss will have different scales for small and large boxes even if their relative errors are similar, whereas the gIoU loss is scale-invariant.

We calculate the ℓ_1 loss as:

$$\mathcal{L}_{\ell_1}(b_i^j, \hat{b}_i^{\sigma^*(j)}) = \|b_i^j - \hat{b}_i^{\sigma^*(j)}\|_1. \quad (13)$$

The gIoU loss is calculated as (we denote the area with set operations for the sake of argument):

$$\mathcal{L}_{\text{iou}}(b_i^j, \hat{b}_i^{\sigma^*(j)}) = 1 - \left(\frac{|b_i^j| \cap |\hat{b}_i^{\sigma^*(j)}|}{|b_i^j| \cup |\hat{b}_i^{\sigma^*(j)}|} - \frac{|B(b_i^j, \hat{b}_i^{\sigma^*(j)})| \setminus (|b_i^j| \cup |\hat{b}_i^{\sigma^*(j)}|)}{|B(b_i^j, \hat{b}_i^{\sigma^*(j)})|} \right), \quad (14)$$

where $|\cdot|$ represents the bounding box area, and the symbols \cup , \cap , and \setminus calculate the area of union, intersection, and subtraction of the two bounding box areas, respectively. $B(b_i, \hat{b}_{\sigma(i)})$ denotes the smallest box enclosing b_i and $\hat{b}_{\sigma(i)}$. The areas are computed by taking the minimum or maximum value of the linear functions of the box coordinates.

3. Experiments

The SVOL dataset is curated upon the ImageNet-VID dataset [22] and three different sketch datasets with varying levels of abstraction: **Sketchy** [6] (least abstract), **TU-Berlin** [10], and **QuickDraw** [24] (most abstract) (see Fig. 2).

3.1. Implementation Details

We uniformly sample 32 frames from a video ($L = 32$), scaled them to 224×224 dimensions, and use them as an input $\mathcal{V} \in \mathbb{R}^{32 \times 3 \times 224 \times 224}$ (3 for RGB channels). Likewise, a sketch is rescaled to 224×224 size, and used as an input $\mathcal{S} \in \mathbb{R}^{224 \times 224}$. The number of CMT layers is set to two (*i.e.*, $l = 2$), and we use 10 object tokens per frame ($M = 10$), a total of 320 object tokens ($N = 320$). We adopt ResNet-50

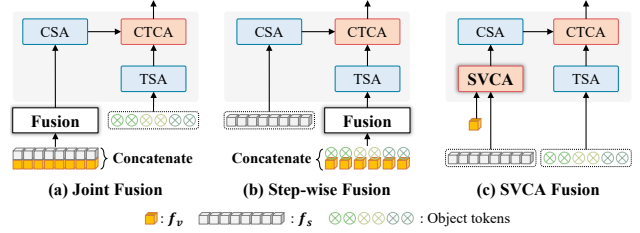


Figure 4. **Three instantiations of sketch-video fusion** contextualize video and sketch information in different ways: **(a) joint fusion:** copy f_s by the size of f_v , concatenate, and fuse them via MLP. **(b) step-wise fusion:** copy f_s by the number of object tokens, concatenate, and fuse them via MLP. The sketch and video representations are later fused through CTCA. **(c) SVCA fusion (ours):** fuse f_s and f_v with SVCA.

and ResNet-18 [7] pre-trained on ImageNet [22] as our video and sketch backbone, respectively.

Due to excessive number of video-sketch pairs, we use an iteration-based batch sampler and randomly sampled a subset from all possible pairings for training. SVANet is trained using AdamW optimizer [18] with an initial learning rate of 10^{-4} and weight decay of 10^{-4} for a batch size of 16. The overall loss weights $\lambda_{L1} : \lambda_{\text{giou}} : \lambda_{\text{cls}}$ are set to 5 : 1 : 2 throughout training. We set different learning schedules (number of iterations and learning rate decay steps) for each sketch dataset as below since their sizes are different.

Settings	Sketchy	TU-Berlin	QuickDraw
# pairs (train)	1,545,801	215,040	2,958,400
Iterations	50,000	20,000	100,000
LR decay step*	30,000	6,000	30,000

*LR is linearly decayed by a factor of 10 at every LR decay step.

3.2. Experimental Setups

Evaluation metrics. We adopt two evaluation metrics for SVOL: 1) \mathbf{R}_μ^k denotes the percentage of samples that have at least one correct result in top- k retrieved results, *i.e.*, Recall@ k , where the correct results indicate that IoU with ground truth is larger than the threshold μ . (we specifically use $k = 1, 5$ and $\mu = 0.5, 0.7$); 2) **mIoU** averages the IoU between predicted boxes and ground truth boxes over all testing samples to compare the overall performance.

Baselines. We set image-level sketch object localization approaches [21, 27] as the SVOL baselines. We find significant room for improvement as they were designed to be conditioned on a single frame rather than an entire video sequence. In addition, we present several instantiations of sketch-video fusion on SVANet, as shown in Fig. 4, and compare them with our final model.

3.3. Comparative Study

We benchmark the model performance on the SVOL task using three different sketch datasets. The results are shown in Table 1. Since image-level baselines (CMA, Sketch-DETR) make predictions at each frame, they neglect the global video context. In contrast, SVANet not only considers

Method (backbone)	Sketchy					TU-Berlin					QuickDraw					ALL (S U T U Q)				
	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU
CMA [†] [27] (R50)	23.18	14.89	39.76	21.29	19.76	20.25	14.55	36.87	20.80	18.64	22.69	15.53	39.86	20.80	21.52	18.89	11.41	33.53	17.23	16.14
Sketch-DETR [†] [21] (R50)	28.78	18.56	46.65	26.50	26.09	30.75	18.97	47.76	27.54	26.24	31.10	19.47	49.39	31.05	28.59	28.23	16.74	44.21	25.54	24.30
SVANet/joint (R50)	33.86	22.56	52.84	30.57	31.46	31.14	19.48	50.17	28.21	29.38	33.95	20.12	54.77	34.11	31.89	29.53	16.50	47.90	27.18	28.59
SVANet/step-wise (R50)	33.31	22.81	53.00	31.07	30.29	30.23	18.19	50.66	27.20	30.41	32.05	21.17	56.34	35.39	31.98	29.64	17.61	48.21	26.29	27.99
SVANet (S3D [31])	32.89	21.11	48.08	27.07	30.83	29.43	18.25	45.72	24.34	28.00	30.86	19.24	46.80	25.48	29.37	27.88	17.26	43.33	22.86	27.87
SVANet (R50 [7])	35.60	23.19	54.06	32.95	33.76	32.10	19.60	51.61	30.94	30.89	34.47	22.30	58.13	37.88	33.40	31.80	18.52	51.44	29.90	30.64

Table 1. **Comparison of SVANet with baselines.** SVANet significantly outperforms baselines on three sketch datasets and on combined dataset (ALL), where we use only overlapping categories between three datasets. [†] indicates the re-implementation based on our settings.

Method	Sketchy→TU-Berlin					Sketchy→QuickDraw					Method	Seen →Unseen categories				
	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU		R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU
CMA [†] [27]	38.03	26.20	49.70	39.07	30.97	40.06	29.89	49.83	37.69	32.45	CMA [†] [27]	18.16	6.89	25.52	9.78	13.37
Sketch-DETR [†] [21]	43.49	32.25	51.71	44.39	36.76	46.02	37.87	59.50	45.34	40.21	Sketch-DETR [†] [21]	25.13	13.20	34.87	18.45	22.51
SVANet (Ours)	54.74	46.67	69.90	56.57	49.01	55.03	47.56	72.04	58.87	49.74	SVANet (Ours)	30.13	18.58	41.18	25.51	29.89

(a) The models are trained with Sketchy dataset and evaluated on QuickDraw or TU-Berlin dataset. To solely see the effect of sketch style differences, we use the same video samples in both training and evaluation, and overlapping categories between the two sketch datasets.

(b) The models are trained on 14 categories of the Sketchy dataset and evaluated on the remaining 5 categories: aircraft, bear, cat, cow, and dog.

Table 2. **Transfer evaluation on (a) unseen datasets and (b) unseen categories.**

spatial context but also effectively models temporal information of video, thereby outperforming them by a significant margin in all metrics across three sketch datasets. Especially, SVANet improves mIoU by 29.4%, 17.7%, and 16.8% over Sketch-DETR on Sketchy, TU-Berlin, and QuickDraw. Also, our final SVANet yields the best results among the several model variants (joint, step-wise), implying the effectiveness of our attention-based fusion. Moreover, we use all three sketch datasets as a single set of query sketches (denoted as ALL in Table 1) to see how the model performs when the same category contains sketches of different styles. Overall, model performances are diminished as a result of a greater diversity of sketch samples. However, SVANet shows only 0.25%p mIoU drop compared to the results on TU-Berlin, which means that SVANet is quite robust to sketch style variations. Lastly, we compare two backbones for video encoding: 3D CNN (S3D [31]) vs. 2D CNN (ResNet50 [7]). We expect the more sophisticated 3D CNN to work better, but 2D CNN outperformed 3D CNN. This shows that CMT can supplant the temporal modeling capability of 3D CNN.

3.4. Transfer Evaluation

The prediction space of our SVANet is not limited to a fixed set of categories. By design, it is possible to match even an unseen sketch to the most similar object by comparing feature-level similarity. For SVOL system to be more practical in real-world applications, they should be able to operate well even with sketches of various shapes and styles. In addition, there should be no constraint that operate only for limited categories, such as object detectors. To this end, we devise two transfer tasks to evaluate the generalization capability of the SVOL systems in two aspects: (i) dataset-level transfer and (ii) category-level transfer.

Formally, we define the transfer evaluation setup as fol-

lows. Let \mathcal{V} , $\{\mathcal{S}_A, \mathcal{S}_B\}$, and $\{\mathcal{C}_A, \mathcal{C}_B\}$ be a video dataset, sketch datasets, and sets of categories in which \mathcal{S}_A and \mathcal{S}_B overlap with \mathcal{V} , respectively. For dataset-level transfer task, we train the SVOL model on \mathcal{V} and \mathcal{S}_A , and evaluate on \mathcal{V} and \mathcal{S}_B , only for categories $\mathcal{C}_A \cap \mathcal{C}_B$. For category-level transfer task, we first split a sketch dataset \mathcal{S}_A into two subsets: \mathcal{S}_A^1 and \mathcal{S}_A^2 , where they are mutually exclusive w.r.t. categories, *i.e.*, $\mathcal{C}_A^1 \cap \mathcal{C}_A^2 = \emptyset$. Then, we train the SVOL model on \mathcal{V} and \mathcal{S}_A^1 , and evaluate on \mathcal{V} and \mathcal{S}_A^2 . We note that there can be more variations to evaluate the transferability of the SVOL system.

Transfer to unseen dataset. We study the transferability of the SVOL models across the sketch datasets with style differences (*e.g.*, line thickness, abstraction degree, *etc.*). The models are trained with Sketchy dataset and evaluated on QuickDraw or TU-Berlin dataset. To solely examine the transferability on unseen datasets, we use the same video samples in both training and evaluation, and overlapping categories between the two sketch datasets. The results are shown in Table 2a. The overall transfer performances across the datasets is much higher than the performance of models that are solely trained on dataset itself (Table 1), since transfer settings use the same video set as in training. We observe that SVANet significantly outperforms baselines in dataset-level transfer, indicating that it effectively learns class-discriminative features independent of sketch style differences. Meanwhile, we expected transfer to TU-Berlin to show better results than transfer to QuickDraw as TU-Berlin appears to be closer to Sketchy than QuickDraw in terms of visual similarity. Contrary to our expectation, transfer to QuickDraw shows better results than transfer to TU-Berlin. We understand this is because the system constructs categorical embedding space by matching the query sketch and video objects based on the key features (*e.g.*, cat’s whiskers,

def. CSA TSA	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	layers	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	p fsm	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU
✓	32.98	19.76	51.74	31.19	30.72	1	33.53	19.95	47.05	24.17	32.30	✗	18.36	6.78	33.54	13.64	22.34
✓ ✓	34.39	21.85	53.69	32.28	32.52	2	35.60	23.19	54.06	32.95	33.76	✓	35.60	23.19	54.06	32.95	33.76
✓ ✓	33.83	20.89	52.29	31.73	31.69	3	35.14	20.18	56.29	32.99	32.77	△	+17.24	+16.41	+20.52	+19.31	+11.48
✓ ✓ ✓	35.60	23.19	54.06	32.95	33.76	4	35.20	22.77	58.34	37.39	33.28		△: performance gain.				

(a) CMT attention operations. default: SVCA + CTCA.

(b) CMT depth.

(c) Per-frame set matching (p fsm).

tokens	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	obj	ℓ ₁	gIoU	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU	frames	R _{0.5} ¹	R _{0.7} ¹	R _{0.5} ⁵	R _{0.7} ⁵	mIoU
5	34.53	20.50	47.61	25.21	32.34	✓	✓		33.64	20.75	46.40	24.37	32.44	16	32.66	18.79	54.24	31.85	31.25
10	35.60	23.19	54.06	32.95	33.76	✓		✓	32.13	18.16	42.32	23.21	31.72	32	35.60	23.19	54.06	32.95	33.76
15	33.89	22.98	54.98	32.64	32.69	✓	✓	✓	35.60	23.19	54.06	32.95	33.76	64	34.91	22.83	53.72	31.23	32.48

(d) Object tokens per frame.

(e) Loss. obj loss is set as default as it is essential.

(f) Input video density.

Table 3. Ablative experiments. Our settings are marked in gray. All experiments are conducted on the Sketchy dataset.

rabbit’s ears, etc.), rather than merely comparing their overall visual appearance. The results suggest that implicit similarity such as symbolic representation of sketches are more important for accurate object localization than explicit similarity such as line thickness or proportion.

Transfer to unseen categories. In Table 2b, we evaluate transferability of SVOL models at the category-level. We use 14 categories of the Sketchy dataset for training, and the remaining 5 categories (aircraft, bear, cat, cow, dog) for evaluation. Compared to the results in Table 1, we observe that SVANet degrades 3.87%p in mIoU since it has never learned which video objects to match the query sketch with. Despite this, SVANet outperforms the baselines when evaluated on unseen categories, implying that SVANet has learned more generalizable representations that can reason about the implicit similarities between sketches and video objects. This enables SVANet to closely embed sketches of the same category in the feature space.

3.5. Ablative Study

CMT attention operations. We study the effect of four attention operations of CMT in Table 3a. Here, SVCA and CTCA are set as default since they are indispensable for making predictions in our design. Each is responsible for modeling interaction between sketch and video, and transforming object tokens into predictions conditioned on the sketch-video joint representations. CSA models the global context of the input sequence and TSA models relationships between object tokens. The default setting work fairly well (mIoU=30.72%), yet SVANet shows better performance with the addition of CSA (+2.33%p) or TSA (+0.97%p). In particular, CSA plays a crucial role in object localization in video since it is in charge of temporal modeling, thus leading to a substantial performance increase. We confirm that all CMT components operate collaboratively on the SVOL task, as they achieve the best performance when used together.

CMT depth. We examine the effect of varying the CMT depth (i.e., number of layers) in Table 3b. A single layer of CMT does not provide sufficient contextualization, resulting in poor R⁵ performance. The overall performance seems

balanced between two to four layers. For R⁵ metric, the deeper the layer, the better the performance, and the best performance is achieved with four CMT layers. However, for more strict R¹ and mIoU metrics, two layers perform the best. Therefore, we make two layers as our default setting.

Per-frame set matching. A straightforward way for training SVANet is to match all predictions with all ground truths as a whole. Although simple, it requires learning all N object tokens simultaneously, regardless of frame order. On the contrary, our per-frame set matching strategy divides N object tokens into L subsets of M object tokens, then matches only a subset to ground truths of its corresponding frame. Although set matching is performed frame-by-frame, SVANet can still make predictions in parallel. We compare our strategy to the straightforward approach in Table 3c. Overall, using per-frame set matching resulted in a significant performance improvement. We see this is because our strategy not only eases optimization by reducing the set matching complexity, but also brings a strong positional inductive bias for object tokens (see empirical evidence in Fig. 6c).

Number of object tokens. In order to see the effect of the number of object tokens used in the CMT layers, we varied their number in Table 3d. Too few tokens (=5) limit sufficient interactions between foregrounds and backgrounds (i.e., No Object), resulting in poor performance, especially for R⁵ metric. On the other hand, too many tokens (=15) diminish performance by producing unnecessary backgrounds. Having 10 object tokens per frame provides a good balance between foreground and background, resulting in a good performance. As we utilize a per-frame set matching strategy, we set the number of object tokens per frame to 10 for the entire 32 frames, thus using a total of 320 object tokens.

Loss components. In Table 3e, we toggle the loss components on and off to understand their impact on training. The objectness loss is used in all cases since it is essential to determine whether a prediction contains the target object. When either ℓ₁ or gIoU [20] loss is disabled, performances drop drastically, especially in R⁵ metric. This implies that both losses are not only important for accurate box localiza-

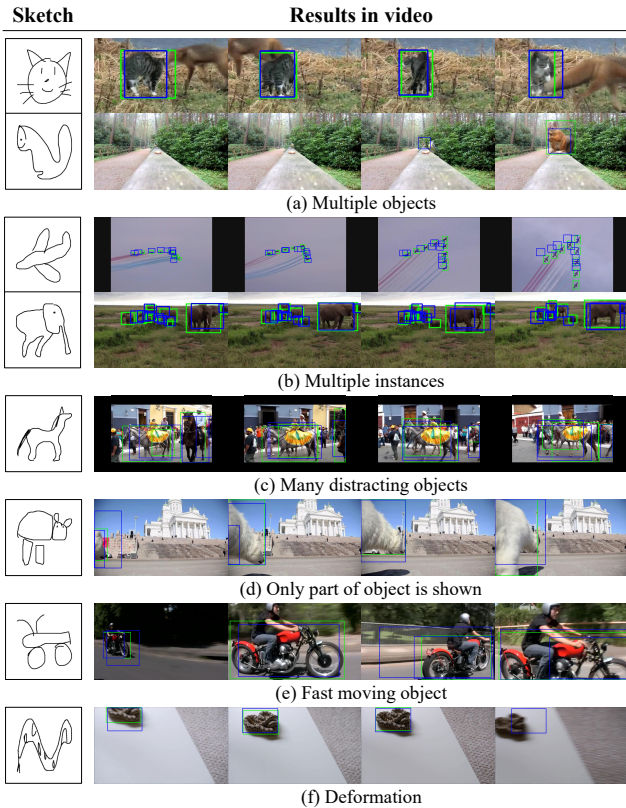


Figure 5. **Qualitative results** of SVANet on QuickDraw dataset. Green and blue boxes represent ground truths and predictions, respectively. SVANet performs well in various challenging scenarios, including: (a) when there are confusable objects; (b) multiple object instances appear in a video; (c) there are many distracting objects; (d) only part of the object is visible; (e) the target object moves quickly; (f) the appearance of the target object is not similar to query sketch.

tion, but also for performing overall predictions well. As we obtain the best results when using all three losses, we confirm that scale-sensitive ℓ_1 and scale-invariant gIoU losses operate complementarily with each other.

Sampling density of video frames. In Table 3f, we study the effect of frame sampling density on input video. We uniformly sample a fixed number of frames across the video and use them as an input to SVANet. By default, we use 32 frames. Compared to the baseline, 16 frames show a particularly sharp performance drop on the strictest metric $R_{0.7}^1$, and 64 frames show overall sub-optimal performance. This is because sparse sampling enables faster processing with less memory, but can easily miss important details since it provides less information. In contrast, dense sampling provides more information, but if the motion of objects is not large, it can be redundant and rather hinder optimization. Here, we study only simple uniform sampling, but different means of sampling may achieve different results.

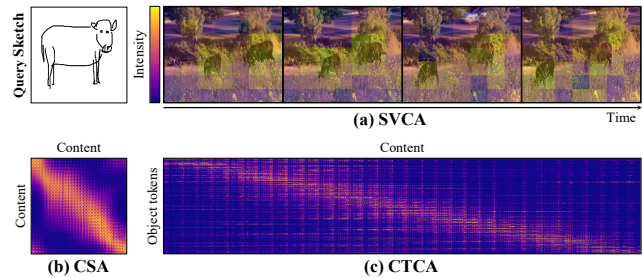


Figure 6. **Attention visualization.** The brighter (yellowish) the color, the higher the attention intensity. Given a query sketch, (a) SVCA mainly attends to regions where the target objects are located; (b) CSA gives a higher attention weight to temporally neighboring contents; (b) CTCA learns which content object tokens should mainly focus on temporally.

3.6. Qualitative Analysis

SVOL results. We present qualitative results of SVANet in Fig. 5 to illustrate how it works in practice. Our system successfully recognizes the objects that correspond to the query sketch and accurately localizes their bounding boxes in a variety of challenging conditions. SVANet works well even when: (a) there are two confusable objects; (b) multiple object instances appear in a video; (c) there are lots of distracting objects; (d) only part of the object is appearing; (e) the target object moves quickly; (f) the appearance of the query sketch is not similar to that of the target object.

CMT attention visualization. In order to understand the behavior of CMT, we visualize its attention maps in Fig. 6. Our observations are as follows: (a) SVCA learns *where to look*, as such, the highlighted area on the attention map aligns well with the actual locations of the sketch object. (b) CSA learns deeper correlation between temporally adjacent sequences when modeling temporal context. (c) CTCA learns *when to look*, thereby giving temporal inductive bias for object tokens in conjunction with per-frame set matching.

4. Conclusion

We introduce a new challenging task termed Sketch-based Video Object Localization (SVOL), where the goal is to localize objects in a video that match a given query sketch. To tackle this task, we propose a strong baseline model named SVANet, which considers the temporal context of video and bridges the domain gap between sketches and videos. SVANet utilizes two key designs to solve the SVOL task as a set prediction problem: a Cross-modal Transformer and per-frame set matching. In our experiments on a newly curated SVOL dataset, we found that SVANet outperforms image-level methods by significant margins. We also conduct comprehensive ablations and show visualizations to analyze the behavior of SVANet. Last but not least, we found that SVANet generalizes well to unseen datasets and novel categories, implying its scalability in real-world scenarios.

References

- [1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing Moments in Video With Natural Language. In *ICCV*, pages 5803–5812, 2017. [1](#)
- [2] Ayan Kumar Bhunia, Viswanatha Reddy Gajjala, Subhadeep Koley, Rohit Kundu, Aneeshan Sain, Tao Xiang, and Yi-Zhe Song. Doodle It Yourself: Class Incremental Learning by Drawing a Few Sketches. In *CVPR*, pages 2293–2302, 2022. [1](#)
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection With Transformers. In *ECCV*, pages 213–229, 2020. [2](#)
- [4] Wengling Chen and James Hays. Sketchygan: Towards Diverse and Realistic Sketch to Image Synthesis. In *CVPR*, pages 9416–9425, 2018. [1](#)
- [5] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer Tracking. In *CVPR*, pages 8126–8135, 2021. [1](#)
- [6] Mathias Eitz, James Hays, and Marc Alexa. How Do Humans Sketch Objects? *ACM TOG*, pages 1–10, 2012. [2](#), [5](#)
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016. [3](#), [5](#), [6](#)
- [8] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-Shot Object Detection With Co-Attention and Co-Excitation. In *NeurIPS*, 2019. [1](#)
- [9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-Image Translation With Conditional Adversarial Networks. In *CVPR*, pages 1125–1134, 2017. [1](#)
- [10] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The Quick, Draw!-AI Experiment. *Mount View, CA, accessed Feb*, page 4, 2016. [2](#), [5](#)
- [11] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What Are You Talking About? Text-to-Image Coreference. In *CVPR*, pages 3558–3565, 2014. [1](#)
- [12] Harold W Kuhn. The Hungarian Method for the Assignment Problem. *Naval research logistics quarterly*, pages 83–97, 1955. [4](#)
- [13] Kin Chung Kwan and Hongbo Fu. Mobi3dsketch: 3D Sketching in Mobile AR. In *CHI*, pages 1–11, 2019. [1](#)
- [14] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High Performance Visual Tracking With Siamese Region Proposal Network. In *CVPR*, pages 8971–8980, 2018. [1](#)
- [15] Pengpeng Liang, Yifan Wu, Hu Lu, Liming Wang, Chunyuan Liao, and Haibin Ling. Planar Object Tracking in the Wild: A Benchmark. In *ICRA*, pages 651–658. IEEE, 2018. [1](#)
- [16] Ioana Literat. “A Pencil for Your Thoughts”: Participatory Drawing as a Visual Research Method With Children and Youth. *International Journal of Qualitative Methods*, 12(1):84–98, 2013. [1](#)
- [17] Jialu Liu. Image Retrieval Based on Bag-of-Words Model. *arXiv preprint arXiv:1304.5168*, 2013. [1](#)
- [18] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*, 2017. [5](#)
- [19] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep Sketch-Based Face Image Editing. *arXiv preprint arXiv:1804.08972*, 2018. [1](#)
- [20] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *CVPR*, pages 658–666, 2019. [5](#), [7](#)
- [21] Pau Riba, Sounak Dey, Ali Furkan Biten, and Josep Lladós. Localizing Infinity-Shaped Fishes: Sketch-Guided Object Localization in the Wild. *arXiv preprint arXiv:2109.11874*, 2021. [1](#), [2](#), [5](#), [6](#)
- [22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet Large Scale Visual Recognition Challenge. *IJCV*, pages 211–252, 2015. [2](#), [5](#)
- [23] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards Style-Agnostic Sketch-Based Image Retrieval. In *CVPR*, pages 8504–8513, 2021. [2](#)
- [24] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The Sketchy Database: Learning To Retrieve Badly Drawn Bunnies. *ACM TOG*, pages 1–12, 2016. [2](#), [5](#)
- [25] Rui Su, Qian Yu, and Dong Xu. Stygbert: A Visual-Linguistic Transformer Based Framework for Spatio-Temporal Video Grounding. In *ICCV*, pages 1533–1542, 2021. [1](#)
- [26] Paul Taele, Rachel Blagojevic, Tracy Hammond, Samantha Ray, Josh Cherian, and Jung In Koh. Sketchrec 2023: 3rd Workshop on Sketch Recognition. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 1–1, 2023. [1](#)
- [27] Aditay Tripathi, Rajath R Dani, Anand Mishra, and Anirban Chakraborty. Sketch-Guided Object Localization in Natural Images. In *ECCV*, pages 532–547, 2020. [1](#), [5](#), [6](#)
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *NeurIPS*, pages 5998–6008, 2017. [2](#)
- [29] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch Your Own Gan. In *ICCV*, pages 14050–14060, 2021. [1](#)
- [30] Sangmin Woo, Jinyoung Park, Inyong Koo, Sumin Lee, Minki Jeong, and Changick Kim. Explore and Match: End-to-End Video Grounding With Transformer. *arXiv preprint arXiv:2201.10168*, 2022. [1](#)
- [31] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-Offs in Video Classification. In *ECCV*, pages 305–321, 2018. [6](#)
- [32] Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep Learning for Free-Hand Sketch: A Survey. *TPAMI*, 2022. [1](#)
- [33] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. LAVT: Language-Aware Vision Transformer for Referring Image Segmentation. In *CVPR*, pages 18155–18165, 2022. [1](#)

- [34] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch Me That Shoe. In *CVPR*, pages 799–807, 2016. [1](#)