DLO Perceiver: Grounding Large Language Model for Deformable Linear Objects Perception

Alessio Caporali, Kevin Galassi, and Gianluca Palli, *Senior Member, IEEE*

***Abstract*—The perception of Deformable Linear Objects (DLOs) is a challenging task due to their complex and ambiguous appearance, lack of discernible features, typically small sizes, and deformability. Despite these challenges, achieving a robust and effective segmentation of DLOs is crucial to introduce robots into environments where they are currently underrepresented, such as domestic and complex industrial settings. In this context, the integration of language-based inputs can simplify the perception task while also enabling the possibility of introducing robots as human companions. Therefore, this letter proposes a novel architecture for the perception of DLOs, wherein the input image is augmented with a text-based prompt guiding the segmentation of the target DLO. After encoding the image and text separately, a Perceiver-inspired structure is exploited to compress the concatenated data into transformer layers and generate the output mask from a latent vector representation. The method is experimentally evaluated on real-world images of DLOs like electrical cables and ropes, validating its efficacy and efficiency in real practical scenarios.**

***Index Terms*—Deformable linear objects, multi-modal models, robotic perception, text-based segmentation.**

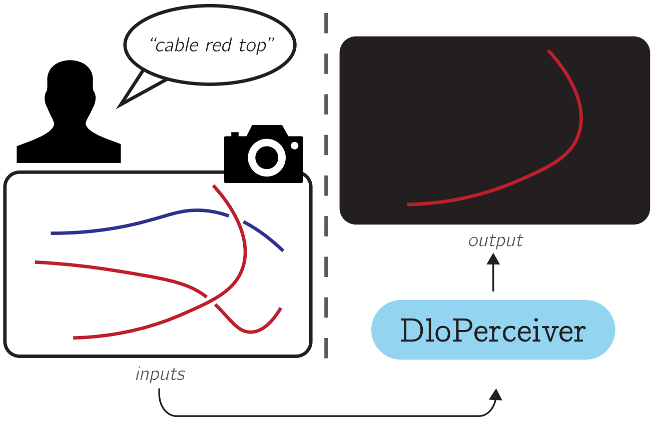


Fig. 1. Text-based image segmentation for DLOs perception.

I. Introduction

In recent years, the rapid advancement of artificial intelligence and computer vision technologies has significantly transformed various fields, including object detection and recognition [1].

Among the objects in the visual domain, Deformable Linear Objects (DLOs) present unique challenges due to their flexible and elongated nature. Indeed, objects such as electrical cables, ropes, and wires, categorized as DLOs, exhibit notable variations in both shape and appearance, thereby posing considerable obstacles for standard perception algorithms [2], [3].

Traditionally, methods for DLO detection often rely solely on visual cues extracted from images [4], [5], [6], neglecting valuable contextual information that may be encoded within associated text descriptions of the scene or task. Indeed, the fusion of visual and textual modalities has shown promise in enhancing the robustness and accuracy of perception systems in several robotic-related fields [7], [8], [9].

Inspired by recent advancements in transformer architectures [1], particularly in natural language processing and computer vision domains, this letter proposes a novel framework that integrates visual and text encoders in a compact and efficient attention-based processing pipeline to achieve DLOs perception. At its core, the idea is to exploit complementary information from both vision and text modalities to segment only the target DLO. In other words, the text-based prompt guides the segmentation of the specific DLO instance of interest. An overview of the proposed text-based DLOs segmentation approach is provided in Fig. 1.

First, both the image and text prompt undergo separate embedding processes through specific encoders. The proposed architecture then employs a Perceiver-inspired structure [10] to compress the encoded data using transformer layers. This process generates the output mask from a learned latent vector representation. Additionally, an auxiliary section of the model functions as a training regularizer. This regularizer aligns associated images and text prompts more closely within a specific latent space using a contrastive learning objective. During inference, the distance in this latent space between the image and the provided text prompt can be used as a measure of similarity, thereby validating the context of the prompt given the available image.

The experimental validation includes real-world images of DLOs, such as electrical cables and ropes, which are common in many domains. The efficacy of combining visual and textual modalities for DLO segmentation tasks is thoroughly investigated, specifically examining the impact of incomplete or incorrect prompts. Additionally, the approach is compared with state-of-the-art methods for DLO instance segmentation and

Received 14 June 2024; accepted 16 October 2024. Date of publication 4 November 2024; date of current version 11 November 2024. This article was recommended for publication by Associate Editor Zhe Min and Editor Cesar Cadena Lerma upon evaluation of the reviewers' comments. This work was supported by the Horizon Europe project *IntelliMan - AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics* under Grant 101070136. *(Corresponding author: Alessio Caporali.)*

The authors are with the DEI - Department of Electrical, Electronic and Information Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: alessio.caporali@unibo.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3491428, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3491428

multi-modal techniques. Finally, a timing and model complexity analysis is reported to better characterize the computational footprint of the proposed method.

In summary, the contributions of this letter can be listed as follows:

1) A novel task for DLOs perception is proposed, leveraging both image and text inputs, and allowing the exclusive segmentation of the target DLO.
2) Efficient Perceiver-inspired architecture for effective compression of encoded visual and textual data using transformer layers, enabling accurate output mask generation from a learned latent vector representation.
3) Exploitation of a contrastive learning objective between image and text to regularize training and provide a similarity score, thereby validating the match between the image and the text prompt during inference.
4) Extensive experimental evaluations and comparisons of the proposed method are reported, including a comprehensive analysis of the effects of correct, partial, and incorrect text-based inputs across a real-world dataset of both cables and ropes.

The associated datasets and source code implementation have been made publicly available on https://github.com/lar-unibo/dlo_perceiver.

The remainder of this letter is organized as follows: Section II provides an overview of the current literature concerning existing visual grounding approaches and perception methodologies for DLOs; Section III presents the network architecture employed to obtain the text-based segmentation of DLOs; Section IV describes the generation and structure of the dataset as well as the training process; Section V outlines the experimental validation employed, featuring an in-depth discussion of the results; Section VI provides concluding remarks and future research directions.

II. Related Works

A. Visual Grounding

Visual grounding is the process of associating linguistic descriptions with their visual counterparts by localizing them in an image [8]. This can be achieved through two main methods: bounding boxes, addressing the Referring Expression Comprehension (REC) task, or segmentation masks, addressing the more complex Referring Image Segmentation (RIS) task, which is also the focus of this letter.

RIS applications appear in various fields, such as visual grounding for navigation, where masks of goal points are provided for a self-driving agent [11], [12], [13], and robotic manipulation tasks, where RIS is augmented with human gaze to indicate the target location of an object, reducing uncertainty from under-specified instructions [14].

Given that visual grounding integrates diverse input sources like vision and language, recent literature has introduced novel approaches and architectures to address this challenge. *CLIP* [7] connects text with images by aligning their embedded spaces, and *ClipSEG* [8] builds on this by using pre-trained image and text encoders from *CLIP*, along with a specialized decoder module to interpret the combined information.

The Perceiver model [10] manages various input configurations using a single transformer-based architecture in a domain-agnostic setting. It employs an asymmetric attention mechanism to condense high-dimensional inputs into a compact latent space, reducing model complexity. This architecture has been successfully applied in several robotic tasks, including a language-conditioned behavior-cloning agent [9].

B. DLOs Perception

Recently, there has been a notable increase in interest in DLOs perception due to its importance for subsequent manipulation tasks [15]. DLOs perception is typically achieved through vision-based methods [2], [3], [5], [6] or tactile-based techniques [16], [17]. Vision-based approaches are preferred due to the availability of various sensors and cameras that can be easily integrated into robotic systems [3], [18]. However, tactile sensing is essential in confined spaces and situations involving occlusions where vision-based perception may struggle. This section reviews recent methodologies for vision-based perception of DLOs, categorized into three main approaches: 2D shape estimation techniques, 3D shape estimation techniques, and multi-modal segmentation.

1) 2D Shape Estimation: Research on estimating the 2D shape of DLOs has focused on data-driven methods for semantic segmentation. Various off-the-shelf deep learning models have been used for this task, including UNet [19], FCN [20], and DeepLabV3+ [2], [21]. A custom CNN architecture with an encoder-decoder scheme is proposed in [22].

Comparisons between real-world and synthetic datasets are conducted in [2] and [23], against the *electrical wires dataset* by [21]. These works highlight how synthetic images can be a viable alternative for DLO segmentation. Combining synthetic with real-world images is shown to improve performance compared to using synthetic images alone [2].

Algorithms for 2D shape estimation of DLOs often use semantic segmentation as a preprocessing step, as shown in [5]. Other approaches include simpler methods like color-based techniques [6] and depth-thresholding [24]. Additionally, some methods estimate DLO endpoints, as demonstrated in [25].

After the segmentation stage, a *tracing* or *merging* procedure is typically employed. Tracing iteratively extends the path of a DLO, as shown in [22] and [25]. Merging-based algorithms, like those by [6] and [5], combine smaller DLO segment estimates into a single detection. Some studies, such as [2] and [23], have applied instance segmentation directly for 2D DLO shape estimation, but these methods often perform poorly when multiple DLOs intersect.

2) 3D Shape Estimation: Acquiring the 3D configuration of DLOs is essential for robotic grasping and manipulation tasks [3]. However, direct 3D shape estimation is less explored compared to 2D methods. Typically, depth data is used to map estimated 2D shapes into 3D space [26], [27]. Challenges in 3D shape estimation arise from sensing technology limitations, especially with thin cylindrical objects like DLOs [18]. High-quality cameras are needed for accurate detection, but they face issues such as larger dimensions, higher costs, and operational constraints when mounted on robot end-effectors compared to 2D cameras [3], [18].

Recent efforts in 3D shape estimation for DLOs include strategies to reduce sensor noise in depth measurements [27] and the use of multi-view stereo techniques [3]. However, these approaches are often time-consuming and may be limited to static scenes.

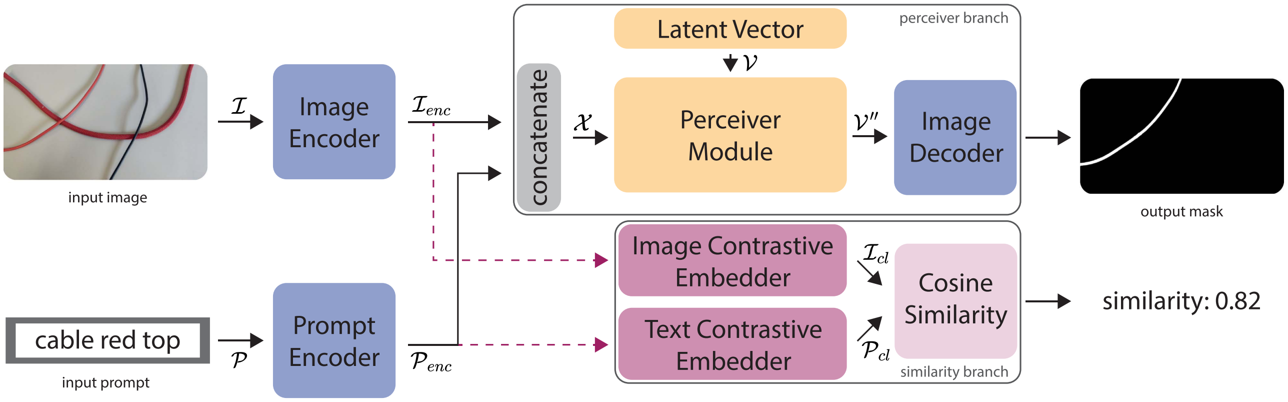


Fig. 2. Network architecture employed for text-based DLO segmentation composed of a perceiver-inspired branch and a contrastive similarity learning branch.

3) Multi-Modal Segmentation: Few works have exploited multi-modal approaches for DLOs segmentation. Notably, the Segment Anything Model (SAM) [28] has been employed in zero-shot settings in [27] and [29]. Both studies use pre-trained models to embed text prompts such as *"ropes"* or *"cables"*. However, in [27], additional handcrafted post-processing steps are required to achieve satisfactory segmentation results, and the method in [29] is tested only in a limited environment.

The proposed method introduces several key improvements over [27] and [29]: 1) It uses more descriptive prompts tailored for specific target-object segmentation rather than general segmentation; 2) It features an efficient, real-time capable architecture in contrast to the full-size foundation models used in the previous works.

III. Method

The proposed *DLOPerceiver* architecture, illustrated in Fig. 2, consists of the following main steps:

- **A) Text and Image Encoding** : Inputs $\mathcal{P}$ and $\mathcal{I}$ are embedded by using specific encoders and concatenated obtaining a new tensor $\mathcal{X}$.
- **B) Perceiver-inspired Processing** : A cross-attention-based processing module is employed to condition a latent image vector $\mathcal{V}$ on the concatenated input tensor $\mathcal{X}$.
- **C) Mask Decoding**: The updated latent image vector $\mathcal{V}''$ is decoded through transposed convolutions and interpolated to restore the original image dimensionality, yielding a logit probability output for each pixel.
- **D) Contrastive Similarity Learning**: Images and text prompts are aligned within a specific latent space through a contrastive learning objective, exploited both during training for regularization and during inference for image/prompt similarity estimation.

The steps mentioned above are analyzed individually in the following section.

A. Text and Image Encoding

The first step in the approach involves embedding the inputs $\mathcal{P}$ (prompt) and $\mathcal{I}$ (image) using specific encoders. The image has dimensions $(h \times w \times 3)$, where h and w are the image height and width respectively. Both $\mathcal{P}$ and $\mathcal{I}$ are embedded such that to have a feature dimension of f.

The text prompt $\mathcal{P}$ is encoded by using a pre-trained model with frozen weights. For this step, BERT [30] is chosen for its proven capabilities but it can be easily replaced with alternative models. The embedded prompt is then passed through a linear layer to reduce the dimensionality of the features to f, obtaining $\mathcal{P}_{\text{enc}} \in \mathbb{R}^f$.

The image $\mathcal{I}$ is processed by using a CNN-based encoder, ResNet101 [31], whose weights are optimized during training to extract features relevant to the DLO perception task considered in this letter. The embedded image $\mathcal{I}_{\text{enc}}$ is initially a tensor of size $(h/4 \times w/4 \times f)$. To facilitate integration with the text input and subsequent processing through the attention-based layers, $\mathcal{I}_{\text{enc}}$ is reshaped to a size $(m \times f)$, where m is the flattened dimension obtained by $(h/4 \times w/4)$.

Thereafter, $\mathcal{P}_{\text{enc}}$ and $\mathcal{I}_{\text{enc}}$ are concatenated to form a single tensor $\mathcal{X} \in \mathbb{R}^{m \times f'}$, where $f' = 2f$. To perform the concatenation, $\mathcal{P}_{\text{enc}}$ is repeated for each pixel of the image, effectively introducing an important conditioning factor for the segmentation task. This dense conditioning approach differs from the standard vector-like embedding methods [8].

Finally, $\mathcal{X}$ is processed by a linear layer to project the dimensionality from f' back to f. Notice that, in this phase, encoding the spatial information of the image pixels is avoided since, in the attention mechanism, rotational encoding is employed instead of standard positional encoding.

B. Perceiver-Inspired Processing

A perceiver-inspired architecture [10] is used to process the image and text concatenated tensor $\mathcal{X}$. The perceiver architecture consists of a series of cross-attention, self-attention, and MLP layers with GEGLU activation functions.

A learnable latent feature tensor $\mathcal{V} \in \mathbb{R}^{n \times f}$ is employed to compress the input data while preserving essential information, thereby reducing the complexity of the network model. Notice that n is in general much smaller than m. This tight latent bottleneck created by $\mathcal{V}$ enables the processing of very large inputs, such as images, while mitigating the quadratic complexity and memory usage typically associated with the attention mechanism.

In particular, two key steps are employed, each one utilizing a processing module that consists of a cross-attention layer, a self-attention layer, and an MLP block. First, the processing module conditions $\mathcal{V}$ on the input tensor $\mathcal{X}$. This is done by using $\mathcal{V}$ as the query and $\mathcal{X}$ as the context in the cross-attention mechanism, resulting in an updated tensor $\mathcal{V}'$. Second, another processing module decodes $\mathcal{V}'$ conditioned on the input tensor $\mathcal{X}$. In this step, $\mathcal{X}$ serves as the query while $\mathcal{V}'$ acts as the context in the cross-attention, yielding the final tensor $\mathcal{V}'' \in \mathbb{R}^{m \times f}$. This approach effectively creates a cross-coupling between the input tensor $\mathcal{X}$ and the latent image tensor $\mathcal{V}$. Compared to the feature-wise transformations in [8] or the direct prompt-based mask decoding in [28], we exploit a perceiver-inspired conditioning of a learned latent image tensor to merge the text prompt with the image data.

C. Mask Decoding

The latent image tensor $\mathcal{V}''$ is reshaped to obtain a tensor of size $(h/4 \times w/4 \times f)$. Then, it goes through a series of transposed convolutions interleaved with batch normalization and ReLU activation functions to reduce the feature dimensionality to 1. Finally, the output is interpolated to the original image size $(h \times w \times 1)$ using bilinear upsampling. This output represents the logit probability of each pixel being the segmentation requested by the prompt.

D. Contrastive Similarity Learning

The similarity branch is a key component of the network model, designed to provide consistent embeddings for the prompt and image tensors. Specifically, it ensures that related latent representations of $\mathcal{I}$ and $\mathcal{P}$ are effectively close in the latent space while unrelated ones are pushed farther away.

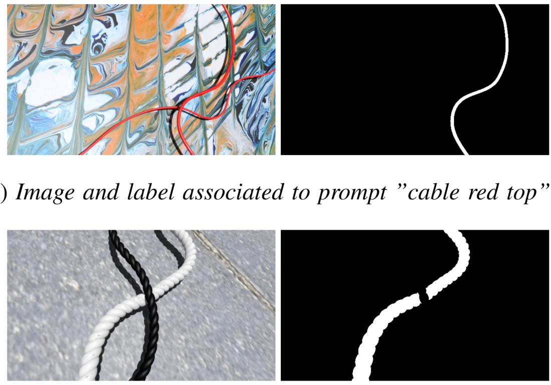
The similarity between these latent representations is evaluated by comparing compressed latent vectors having the same dimensionality. Thus, both $\mathcal{I}_{\text{enc}}$ and $\mathcal{P}_{\text{enc}}$ are transformed to have the same feature size f, i.e. the goal is to obtain $\mathcal{I}_{\text{cl}} \in \mathbb{R}^f$ and $\mathcal{P}_{\text{cl}} \in \mathbb{R}^f$. To achieve this objective, $\mathcal{P}_{\text{enc}}$ is processed using a linear layer that preserves the output dimensionality at f. For the image $\mathcal{I}_{\text{enc}}$, a sequence of convolutional layers interleaved with batch normalization and ReLU activation functions compresses the initial tensor. Finally, a linear layer projects the resulting dimensionality to f. During training, $\mathcal{I}_{\text{cl}}$ and $\mathcal{P}_{\text{cl}}$ are compared by means of a distance-related function, more details are provided in Section IV-B.

Instead, during inference, a measure of similarity $s \in [0, 1]$ can be exploited to better characterize the relationship between the captured image and the supplied prompt. To obtain the similarity score within these bounds, the cosine similarity between the text $\mathcal{P}_{\text{cl}}$ and image $\mathcal{I}_{\text{cl}}$ feature vectors is computed as follows:

$$\text{cos\_sim}(\mathcal{P}_{\text{cl}}, \mathcal{I}_{\text{cl}}) = \frac{\mathcal{P}_{\text{cl}}^T \mathcal{I}_{\text{cl}}}{\|\mathcal{P}_{\text{cl}}\| \, \|\mathcal{I}_{\text{cl}}\|}.$$

Since the output of the cosine similarity lies within the range $[-1, 1]$, the value is rescaled to fit the desired bounds.

Note that other methods, such as the multi-mask output selection in [28], provide confidence estimates for their predictions. In contrast, the proposed similarity value does not reflect prediction confidence but rather tries to assess the 'soundness' of the prompt relative to the image.



(a) *Image and label associated to prompt "cable red top".*

(b) *Image and label associated to prompt "rope white bottom".*

Fig. 3. Dataset samples obtained by 1) rendering synthetic images of cable and rope-like objects, and 2) associating label masks and prompts for a given object instance.

IV. Dataset Generation and Training Process

A. Training Dataset Generation

Following the procedure described in [2], a photorealistic synthetic dataset of DLO images is generated, featuring both cable and rope-like objects. Therefore, a text prompt is associated with each DLO instance in a given image. Examples of cable and rope cases, along with their associated image labels and text prompts, are shown in Fig. 3. Details about the rendering and the association of the text prompt are provided in the following sections.

1) Image Rendering: The rendering of the synthetic images utilizes Blender [32], where a mesh object is created from the generated spline-based DLO model [2]. The DLO mesh is obtained by specifying the object's thickness and color. Additionally, the final appearance of the mesh can be customized to appear either smooth or braided, reflecting different DLO objects: cables and wires for the smooth case, and ropes for the braided case. Finally, an environment map is used to simulate realistic lighting conditions, and a supporting plane with a specific texture is added to the scene. All the mentioned texture, light, and color properties are randomly selected during the generation process to increase the dataset's variance. This approach allows for different combinations of shadows, objects, and background scenes to be simulated, further enhancing the generalization capabilities of the data-driven approaches trained on the generated dataset. Along with each generated image sample, the rendering pipeline provides associated ground-truth data in the form of a label mask, where each pixel is labeled with the corresponding object instance identifier.

2) Image/Prompt Association: For each generated image $\mathcal{I}$, a training sample is created for every DLO instance within the image. Specifically, a text prompt is associated with each DLO instance i as follows:

$$\mathcal{P}_i = < \textit{object color position} > .$$

The *object* attribute is either *"cable"* or *"rope"*, depending on the object type rendered in the scene. The *color* attribute is

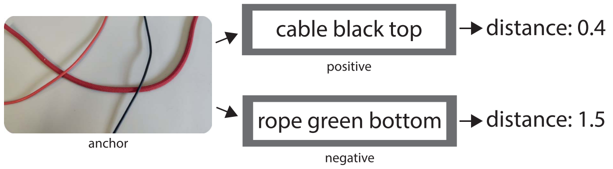


Fig. 4. Schema of triplet loss contrastive learning with an example of positive and negative prompts.

one of *"red," "green," "blue," "black," "white," "gray," "yellow," "orange," "purple"* or *"striped."* The *position* attribute is determined by processing the instance masks of the scene in case of intersection between different DLOs, and is either *"top"* or *"bottom,"* with the top position assigned to the DLO placed at the top of the corresponding intersection area. The results of this process are shown in Fig. 3 for two different DLO instances. Therefore, each dataset sample can be denoted as a tuple $(\mathcal{I}, \mathcal{M}_i, \mathcal{P}_i)$, where $\mathcal{I}$ is the source RGB image, and $\mathcal{M}_i$ and $\mathcal{P}_i$ are the mask and prompt of DLO instance i.

B. Training Process

The network is optimized by employing a loss function composed of two terms: segmentation consistency and image/prompt similarity guidance.

Denoting by $\hat{\mathcal{M}}_i$ the predicted mask of DLO instance i obtained from Section III-C, and by $\mathcal{M}_i$ the ground truth mask, the segmentation loss is simply defined as:

$$L_{\text{seg}} = \text{BinaryCrossEntropyLoss}(\hat{\mathcal{M}}_i, \mathcal{M}_i).$$

Regarding the contrastive similarity branch discussed in Section III-D, a more complex loss computation is required.

The contrastive-based approach utilizes the triplet loss criterion [33] for the branch optimization. This criterion considers a triplet of vectors â commonly denoted as anchor, positive, and negative â and computes the distances between anchor-positive and anchor-negative pairs. Specifically, during training, the goal is to minimize the anchor-positive distance while maximizing the anchor-negative one.

In this context, the image vector $\mathcal{I}_{\text{cl}}$ serves as the anchor, while the positive vector is represented by the text associated with the image sample, i.e. $\mathcal{P}_{\text{cl},i}$. Conversely, the negative vector is randomly generated as a *negative* prompt example $\bar{\mathcal{P}}_{\text{cl},i}$, ensuring that misleading or ambiguous prompts are avoided. An example of anchor, positive and negative samples is provided in Fig. 4. Therefore, the loss of the similarity branch can be denoted as:

$$L_{\text{cl}} = \text{TripletLoss}(\mathcal{I}_{\text{cl}}, \mathcal{P}_{\text{cl},i}, \bar{\mathcal{P}}_{\text{cl},i}).$$

Finally, the combined loss is obtained as a weighted sum of the two losses, i.e. $L = L_{\text{seg}} + \lambda L_{\text{cl}}$ where $\lambda \in \mathbb{R}_+$ is a hyperparameter to select.

V. Experiments

The experiments are performed employing a workstation equipped with an Intel Core i9-10900K CPU and an NVIDIA GeForce RTX 2080Ti GPU. The method is implemented in PyTorch 2.0 and both the training and the inference are performed on a single GPU.

A. Optimization Details

The network, detailed in Section III, is optimized employing a dataset from Section IV composed of about 7000 samples with a resolution of 640×360 pixels. The usual 90–10% split is used to obtain the training and validation datasets.

The dataset is heavily augmented during the learning process to address the reality gap. The augmentation scheme includes hue, saturation, and value randomization; image flipping; image blur with noise; random cropping and padding; grid dropout; and noise feature augmentation [34].

For the image encoder, a ResNet-101 backbone pre-trained on ImageNet is used, which is optimized during training. Conversely, a frozen BERT text encoder [30] is utilized for the prompt.

The perceiver module of Section III-B consists of 1 cross-attention block followed by 3 self-attention blocks. Each block is composed of 4 heads. After each attention operation, two linear layers with GELU activation and dropout are added. Concerning the tensor dimensions, $f = 256$ and $n = 256$ are employed.

The latent image vector is learned through the optimization of the network. It is initialized with a random normal distribution with a mean of 0 and a variance of 0.2.

The network is optimized for $400\,000$ steps with the final weights selected as the ones corresponding to the lowest validation loss. Additional hyper-parameters defined are the batch size of 6, Adam as optimizer, loss factor $\lambda = 0.1$, and a polynomial learning rate adjustment policy with power 0.97 starting from 1×10^{-4} to a minimum of 10^{-7} .

B. Test Dataset and Metrics

To evaluate the performances on real data, a *test set* of 90 manually labeled real images of electrical wires and ropes with varying diameters and collected in different real scenarios is used. The test dataset is organized into 3 categories, each one containing 30 images. The categories are defined as follows:

S1: Scenes with target DLOs placed on a surface without other distracting objects. Challenges include high-contrast shadows, potential chromatic similarities with the background, lighting conditions, and perspective distortions.

S2: Scenes featuring target DLOs over complex and highly detailed backgrounds without other distractors. The algorithm faces the challenge of accurately extracting DLOs from cluttered scenes with chaotic backgrounds.

S3: Scenes depicting target DLOs in realistic settings, such as industrial environments. Challenges arise from metallic surfaces reflecting wires and other distracting objects like commercial electromechanical components typical of such environments.

For each category, the *test set* can be further divided into 3 subcategories, each containing 10 images with only cables, 10 images with only ropes and 10 images with mixed cables and ropes. Alternatively, the *test set* can be divided into 2 subcategories based on the difficulty level: easy with only two DLO instances in the scene, and difficult with more than two DLO instances in the scene. In this case, each subcategory contains 15 images. As detailed in Section IV-A1, cables are defined as DLOs with a smooth and uniform texture, while ropes are distinguished by their braided appearance.

The organization of the *test set* allows to evaluate the network performances in different scenarios and to characterize the

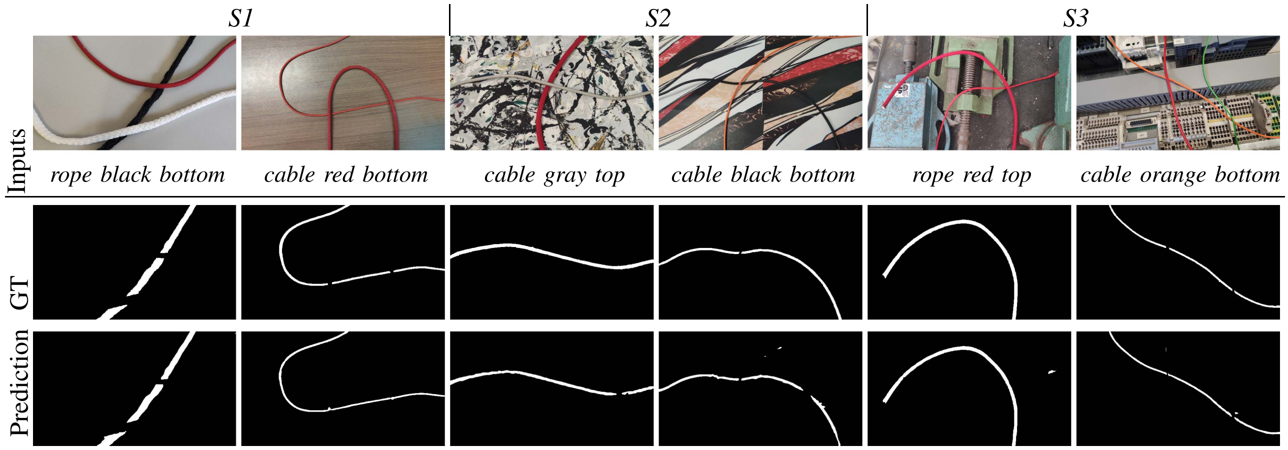


Fig. 5. Qualitative results on the categories *S1*, *S2* and *S3* of the *test set*. The first row shows the input images with the corresponding text prompt. The second row shows the ground truth masks. The third row shows the predicted masks.

TABLE I
QUANTITATIVE RESULTS ON THE *TEST SET* WITH RESPECT TO THE DLOS TYPE, SCENE COMPLEXITY AND THE AVERAGE

Groups		*S1*	*S2*	*S3*
DLOs type	only cables	0.778	0.739	0.776
	only ropes	0.785	0.771	0.812
	mixed	0.706	0.733	0.772
Scene complexity	easy	0.801	0.817	0.809
	difficult	0.727	0.701	0.772
Average		**0.764**	**0.759**	**0.790**

The results are the mean IoU with predictions thresholded at 0.5.

TABLE II
ABLATION EXPERIMENTS SHOWING IOU VALUES ON THE *TEST SET* (*S1*, *S2*, *S3*) FOR DIFFERENT IMAGE ENCODERS, WITH (✔) AND WITHOUT (✗) THE USE OF THE CONTRASTIVE SIMILARITY LEARNING BRANCH DURING TRAINING

Set	SwinT		SwinS		ResNet50		ResNet101	
	✗	✓	✗	✓	✗	✓	✗	✓
S1	0.612	0.705	0.710	0.739	0.649	0.733	0.628	0.764
S2	0.689	0.701	0.731	0.735	0.730	0.741	0.731	0.759
S3	0.693	0.702	0.742	0.707	0.725	0.725	0.760	0.790
AVG	0.665	0.703	0.728	0.727	0.701	0.733	0.706	0.771

generalization capabilities of the model across different real DLO objects, backgrounds, and scene complexities.

Concerning the evaluation metric, following previous works [5], the evaluation is performed by employing the Intersection over Union (IoU) score. Specifically, the network produces a binary mask $\hat{\mathcal{M}}_i$, which corresponds to the predicted segmentation of the target DLO i. The output is compared against the ground truth label $\mathcal{M}_i$ as $\text{IoU} = \frac{|\hat{\mathcal{M}}_i \cap \mathcal{M}_i|}{|\hat{\mathcal{M}}_i| + |\mathcal{M}_i|}$.

C. Segmentation Results

This section presents the results of the segmentation experiments, featuring both quantitative and qualitative analysis.

1) Quantitative Results: Table I reports the quantitative results on the *test set* with respect to the different categories and subcategories detailed in Section V-B. The results are reported in terms of the mean Intersection over Union (mIoU) with predictions thresholded at 0.5. The average scores reported for the different *S1*, *S2* and *S3* categories are computed as the mean of the scores obtained for each image, where the score of an image is the mean IoU across the different DLO/prompts pairs. The other scores, concerning the DLOs type and the scene complexity, are computed as the mean of the scores obtained for each image in the corresponding subcategory. Overall, the network achieves consistent performances across the different objects, backgrounds, and scene complexities. In the *difficult* scene complexity case, the network shows a decrease in performance, especially in the *S2* category. This is due to the presence of more DLO instances in the scene, which makes the task more challenging, as expected.

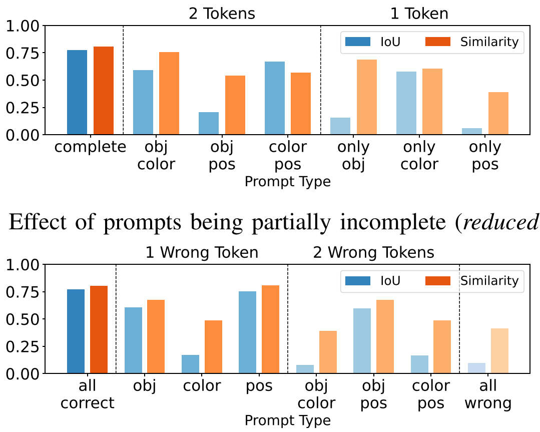
2) Qualitative Results: The quantitative results are supported by the qualitative analysis of Fig. 5. The figure shows the input images with the corresponding text prompts, the ground truth masks and the predicted masks. The network is able to correctly segment the target DLOs in different scenarios, even in the presence of difficult backgrounds and scene complexities.

D. Ablation Studies

To better understand the impact of various components of the architecture, this section analyzes the contribution of the similarity branch (Section III-D) to the overall network segmentation accuracy. Additionally, different backbones for the image encoder (Section III-A) are evaluated. Both results are summarized in Table II where IoU scores across the *test set* categories for different training procedures and backbone models are shown.

1) Effect of Similarity Branch During Training: By comparing the different columns of Table II, it is possible to appreciate the regularization effect of the similarity branch during the training process, that helps to push the image and text prompt in a consistent manner in the latent space, resulting in closer accuracy results across the different *test set* categories.

2) Employing Different Image Encoders: Other backbones tested for comparison against ResNet101 include the smaller ResNet50 and the Swin Transformer architecture [35] in its tiny (SwinT) and small (SwinS) variants. The ResNet and Swin Transformer models have a comparable number of parameters



(a) Effect of prompts being partially incomplete (*reduced*).

(b) Effect of prompts being partially or entirely incorrect (*wrong*).

Fig. 6. Reduced and wrong text prompt analysis.

relative to their respective variants. Table II highlights similar results across the different backbones, with ResNet101, in the contrastive learning configuration, achieving the highest IoU score.

E. Text Prompt Analysis

The text prompt constitutes a fundamental element of the proposed approach. Ideally, it could be supplied by an operator interacting with the robot. Therefore, the possibility of incomplete or incorrect prompts is not negligible. In the following, the effects of *reduced* and *incorrect* prompts are analyzed.

1) Reduced Prompt: As outlined in Section IV-A, a prompt is composed of 3 *tokens* (object, color and position). Thus, to evaluate the reduced prompt settings, for each image/prompt pair in the *test set*, the prompt is modified by removing one or more tokens. The results in terms of accuracy and similarity (see Section III-D) are depicted in Fig. 6(a). The x-axis organizes different configurations into *complete* (all 3 tokens), *2 tokens*, and *1 token* for clarity. Among the tokens, color and object have the most significant impact on accuracy, while the position token has a lesser effect.

2) Wrong Prompt: A similar analysis to the reduced case is conducted for the wrong settings. In this scenario, an entire prompt composed of 3 tokens is always provided, but one, two, or all three tokens are entirely incorrect. The results of this test are illustrated in Fig. 6(b). The x-axis in this case spans from entirely correct to entirely wrong, including intermediate configurations. Consistent with the findings of Section V-E1, color is the most crucial token type, leading to the poorest performance when incorrect values are provided. Importantly, the similarity score appears to reliably reflect the varying degrees of incorrectness.

F. Comparison With State-of-the-Art Methods

In the DLOs perception domain, SOTA methods predominantly focus on the vision-based instance segmentation, i.e. these methods return all DLOs present in the given RGB input. Alternatively, more general multi-modal approaches have emerged, able to utilize both text and image inputs. While the former methods are more closely aligned with the specific domain of application proposed in this research, the latter exhibit greater similarity to the proposed task modality. A comparison between these two groups is shown in Table III, where with *DloPerceiver* the proposed method is addressed.

TABLE III
COMPARISON OF IOU SCORES ON THE *TEST SET* BETWEEN BASELINE INSTANCE SEGMENTATION METHODS FROM THE DLOS DOMAIN AND MULTI-MODAL APPROACHES

Set	DLOs Instance Segmentation						Multi-Modal Models		
	FASTDLO		*RT-DLO*		*mBEST*		*ClipSEG*	*DINO+SAM*	*DLOPerceiver*
	★	†	★	†	★	†	★	★	★
S1	0.537	0.794	0.521	0.776	0.428	0.701	0.264	0.096	0.764
S2	0.455	0.644	0.430	0.611	0.286	0.433	0.143	0.051	**0.759**
S3	0.403	0.614	0.401	0.623	0.252	0.387	0.227	0.016	**0.790**
AVG	0.465	0.684	0.451	0.670	0.322	0.507	0.212	0.055	**0.771**

★ denotes the entire set of images, while † refers to the only cables subgroup.

1) DLOs Instance Segmentation Methods: The comparison is conducted with *FASTDLO* [4], *RT-DLO* [5], and *mBEST* [6]. These methods utilize the same semantic segmentation stage of FASTDLO to obtain a binary mask, followed by their specific instance segmentation approaches. Consequently, all DLOs in the image are retrieved and associated according to the ground truth prompt. It's worth noting that these methods do not utilize textual information for prediction. The poor results of all these methods are mostly due to the segmentation pre-processing step being trained only on cable-like DLOs. Indeed, analyzing the results on the *cable only* subcategory reveals stronger performances with similar accuracy results among the baselines. However, *DloPerceiver* achieves stronger performances, especially on the *S2* and *S3* subcategories.

2) Multi-Modal Approaches: The comparison is conducted with *DINO+SAM*, a version of *SAM* [28] that utilizes *DINO* [36] as a language model to encode textual information. Additionally, *ClipSEG* [8] represents a state-of-the-art textual-image model for segmentation tasks. Throughout the comparison, all models are inferred using the same prompt and input image. The results in Table III reveal poor accuracy for the proposed multi-modal baselines. This underperformance may rise from the larger image reduction (factor of 16) during encoding, compared to the more moderate factor of 4in the proposed encoder, which better preserves thin DLO details. Moreover, general-purpose datasets used to train multi-modal models may not capture DLO-specific features effectively.

G. Timings and Model Complexity

The timing of the proposed method is evaluated using the outlined *test set* and workstation setup. To account for GPU warm-up, an initial run is performed on the entire *test set*, and the average timings are computed during a subsequent run.

Overall, the network processes an image in an average of 23 ± 3 ms. Encoding the text and image takes an average of 11 ms and 7 ms, respectively. The perceiver-based processing module with the similarity branch takes only 5 ms on average. Compared to SOTA models, *DLOPerceiver* runs at approximately 40 Hz, making it faster than DLO instance segmentation methods (e.g., 30 Hz for RT-DLO and 20 Hz for FASTDLO), and an order of magnitude faster than multi-modal models. This improvement is due to the compact size of its encoding modules and the efficiency of the perceiver architecture. Indeed, *DLOPerceiver* comprises only approximately 46 million learnable parameters.

VI. Conclusion

In this work, a learning-based multi-modal method for the perception of Deformable Linear Objects (DLOs) in real-world scenarios is proposed making use of both image and text inputs. The approach involves segmenting target DLOs in images using a perceiver-based architecture. A photorealistic synthetic dataset comprising cables and ropes, the two primary types of DLOs, is employed for model optimization. The method is evaluated on a real-world dataset featuring images of cables and ropes in diverse scenarios. The results demonstrate the method's capability to accurately segment DLOs, even in challenging conditions characterized by complex backgrounds and multiple DLOs instances.

In future work, the proposed approach will be introduced in a collaborative manipulation task targeting the assembly and routing of DLOs. Additionally, there is potential to extend the method to accommodate more complex text-based prompts, such as those comprising multiple sentences or providing more detailed descriptions of the target DLOs.

References

[1] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5625–5644, Aug. 2024.

[2] A. Caporali, M. Pantano, L. Janisch, D. Regulin, G. Palli, and D. Lee, "A weakly supervised semi-automatic image labeling approach for deformable linear objects," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 1013–1020, Feb. 2023.

[3] A. Caporali, K. Galassi, and G. Palli, "Deformable linear objects 3D shape estimation and tracking from multiple 2D views," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3852–3859, Jun. 2023.

[4] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "FASTDLO: Fast deformable linear objects instance segmentation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9075–9082, Oct. 2022.

[5] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, "RT-DLO: Real-time deformable linear objects instance segmentation," *IEEE Trans. Ind. Inform.*, vol. 19, no. 11, pp. 11333–11342, Nov. 2023.

[6] A. Choi, D. Tong, B. Park, D. Terzopoulos, J. Joo, and M. K. Jawed, "mBEST: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 4863–4870, Aug. 2023.

[7] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[8] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7076–7086.

[9] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proc. Conf. Robot Learn.*, 2023, pp. 785–799.

[10] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 4651–4664.

[11] N. Hosomi et al., "Trimodal navigable region segmentation model: Grounding navigation instructions in urban areas," *IEEE Robot. Automat. Lett.*, vol. 9, no. 5, pp. 4162–4169, May 2024.

[12] N. Rufus, K. Jain, U. K. R. Nair, V. Gandhi, and K. M. Krishna, "Grounding linguistic commands to navigable regions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8593–8600.

[13] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi, "Ground then navigate: Language-guided navigation in dynamic scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 4113–4120.

[14] K. Qian, Z. Zhang, W. Song, and J. Liao, "GVGNet: Gaze-directed visual grounding for learning under-specified object referring intention," *IEEE Robot. Automat. Lett.*, vol. 8, no. 9, pp. 5990–5997, Sep. 2023.

[15] A. Caporali, P. Kicki, K. Galassi, R. Zanella, K. Walas, and G. Palli, "Deformable linear objects manipulation with online model parameters estimation," *IEEE Robot. Automat. Lett.*, vol. 9, no. 3, pp. 2598–2605, Mar. 2024.

[16] S. Pirozzi and C. Natale, "Tactile-based manipulation of wires for switchgear assembly," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 6, pp. 2650–2661, Dec. 2018.

[17] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," *Int. J. Robot. Res.*, vol. 40, pp. 1385–1401, 2021.

[18] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, "New metrics for industrial depth sensors evaluation for precise robotic applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5350–5356.

[19] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5687–5694, Apr. 2022.

[20] W. Wu, Y. Zhu, X. Zheng, and Y. Guo, "A novel cable-grasping planner for manipulator based on the operation surface," *Robot. Comput.-Integr. Manuf.*, vol. 73, 2022, Art. no. 102252.

[21] R. Zanella, A. Caporali, K. Tadaka, D. D. Gregorio, and G. Palli, "Auto-generated wires dataset for semantic segmentation with domain-independence," in *Proc. IEEE Int. Conf. Comput. Control Robot.*, 2021, pp. 292–298.

[22] X. Huang, D. Chen, Y. Guo, X. Jiang, and Y. Liu, "Untangling multiple deformable linear objects in unknown quantities with complex backgrounds," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 1, pp. 671–683, Jan. 2024.

[23] J. Dirr, D. Gebauer, J. Yao, and R. Daub, "Automatic image generation pipeline for instance segmentation of deformable linear objects," *Sensors*, MDPI, vol. 23, no. 6, 2023, Art. no. 3013.

[24] A. Keipour, M. Bandari, and S. Schaal, "Deformable one-dimensional object detection for routing and manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4329–4336, Apr. 2022.

[25] D. D. Gregorio, G. Palli, and L. D. Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 662–677.

[26] P. Kicki, A. Szymko, and K. Walas, "DLOFTBs–Fast tracking of deformable linear objects with b-splines," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 7104–7110.

[27] Z. Sun, H. Zhou, L. Nanbo, L. Chen, J. Zhu, and R. B. Fisher, "A robust deformable linear object perception pipeline in 3D: From segmentation to reconstruction," *IEEE Robot. Automat. Lett.*, vol. 9, no. 1, pp. 843–850, Jan. 2024.

[28] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 4015–4026.

[29] S. Kozlovsky, O. Joglekar, and D. D. Castro, "ISCUTE: Instance segmentation of cables using text embedding," 2024, *arXiv:2402.11996*.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[32] M. Denninger et al., "Blenderproc: Reducing the reality gap with photorealistic rendering," in *Proc. 16th Robot.: Sci. Syst., RSS, Workshops*, 2020.

[33] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[34] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales, "A simple feature augmentation for domain generalization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8866–8875.

[35] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.

[36] S. Liu et al., "Grounding DINO: Marrying DINO with grounded pretraining for open-set object detection," 2023, *arXiv:2303.05499*.