LangGas: Introducing Language in Selective Zero-Shot Background Subtraction for Semi-Transparent Gas Leak Detection with a New Dataset

Supplementary Material

7. Methane Release From GasVid

The total amount of methane released during the capture of the GasVid dataset can be calculated using Equation 2, where m_{total} represents the total mass of methane released, n is the number of videos, i denotes the class label corresponding to the flow rate, and m_i is the flow rate of the i-th class in g/h. Each flow rate lasted for 3 minutes. Based on the flow rate data from the GasVid paper, the total methane release is 12906.385g.

$$\begin{aligned} m_{\text{total}} &= n \times \sum_{i=0}^{7} \left(\frac{m_i}{60} \times 3 \right) \\ &= \frac{31 \times 3}{60} \times \sum_{i=0}^{7} m_i \\ &= 1.55 \times 8326.7 \text{ g} \\ &= 12906.385 \text{ g} \end{aligned} \tag{2}$$

8. Further Review of VLMs

CLIP [30] is a remarkable work that has inspired downstream work in segmentation, detection, etc. LSeg [11] adapted CLIP into segmentation by calculating the similarity of the text query with every pixel on the feature map of the image, classifying each pixel into one of the text queries. It then used a special regulation block to decode the feature map into segmentations. This straightforward way has also been used in OwlVit [23] and Owl-V2 [24]. In OwlVit, they pre-trained the CLIP encoder using contrastive loss and transferred the model into detection by removing the pooling operation with a classification and localization head to archive language-guided detection.

Besides the image-text contrastive loss function, align before fuse (ALBEF) [12] also used image-text matching and masked language modelling like in BERT [5]. Their model has an image encoder, a text encoder, and a multimodality encoder.

Although ALBEF was not trained on grounding or localization tasks, their Grad-CAM [36] has shown a strong localization correlation between phrases and text. This is further improved by [8, 53]. Grounding-DINO [19] and GLIP [14], on the other hand, are specifically trained on grounding tasks and trained in object detection fashion by producing bounding boxes for phases. Both the Grad-CAM and the bounding box can be used to prompt a segmentation model such as SAM [9], or SAM 2 [32] to generate language-guided instance segmentation masks like in Grounding-SAM [33] and APOVIS [20].

Another line of work took a generative approach [1, 3, 10, 18, 25, 26, 44, 50]. In these works, GPT-4 serials [25, 26] and llama-like [1, 18] models use pure language as an interface, take in instruction as text prompt and generate

Algorithm 1: Temporal Filtering Algorithm

Input: *current boxes*, *past boxes*, *image size*
Output: *valid boxes*

```
Set valid boxes as an empty list;
for each current box in current boxes do
    if area of current box > image area × ignore large threshold then
        Skip this box;
    Set matched boxes to 0;
    for each past frame boxes in the last maximum past frames do
        Find overlap between current box and past frame boxes;
        Find position difference between current box and past frame boxes;
        if any overlap > IoU match threshold OR all position differences < absolute shift threshold then
            Increase matched boxes by 1;
    if matched boxes ≥ match threshold then
        Add current box to valid boxes;
Set matched boxes as an empty list;
if valid boxes is empty AND past boxes length ¿ 3 then
    for each first frame in the last 3 frames do
        for each second frame in the last 3 frames do
            if first frame == second frame then
                Skip this frame;
            for each box in first frame do
                if any overlap with boxes in second frame > IoU match threshold then
                    Add box in first frame to matched boxes;
return valid boxes;
```

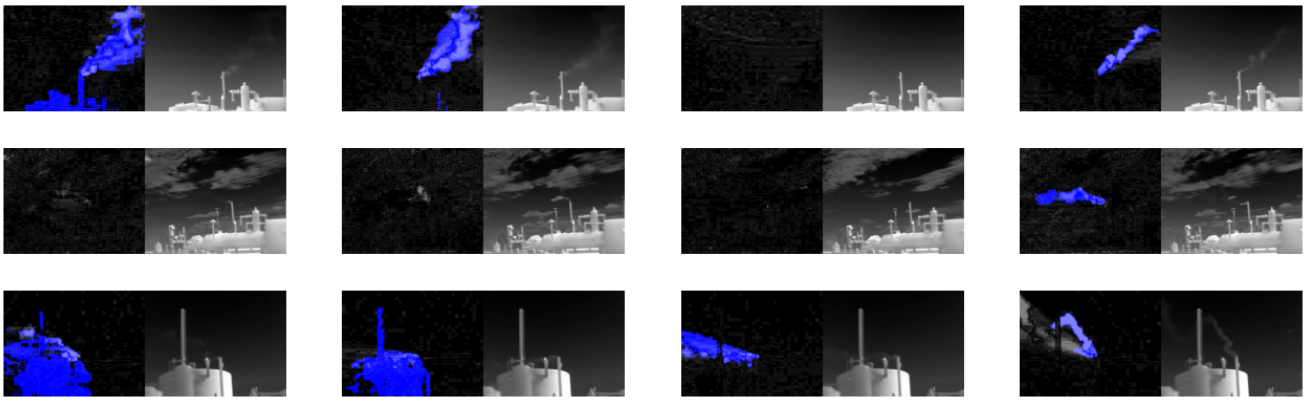

Figure 6. **Selected Samples from GasVid**: The two left columns display failure cases, and the two right columns show successful cases. In each pair, the left image shows the background subtraction result, with blue indicating the segmentation output (artifacts may appear), while the right image is the original frame. The three rows correspond to videos with GasVid IDs 1239, 2570, and 2579, recorded at distances of 12.6m, 15.6m, and 6.9m, respectively.

output as pure text (such as location information in coordination). Florence, on the other hand, uses special tokens for different tasks (such as segmentation, detection, etc) and also uses special tokens for generated results. Some other works [3, 10, 44] also used special tokens for segmentation results.

9. Qualitative Experiments on GasVid

We excluded videos recorded at 18.6 m (following VideoGasNet [46]) and selected examples showing two failure cases and two successful cases, as shown in Figure 6. The experiment used MOG2 as the background subtractor, OWLv2 [24] as the visual language model with a threshold of 0.06, enhancement factor of 10, and both temporal filtering and SAM 2 enabled. The results indicate that the model can localize and segment leakage with reasonable performance, although worse than the synthetic dataset due to real-world noise, artifacts in background subtraction, etc. Future work should be done on how to improve this method on real-world captured videos.

In the success cases, two samples (from the third column and first two rows) are true negatives, showing that noise is not mistakenly segmented as a leak, while the remaining examples are true positives with well-aligned segmentation boundaries. In the sample in the fourth column of the third row, the model avoids an artifact from background subtraction that is not a leak. In the failure cases, the first and third videos show over-segmentation of non-leak objects, and in the second video, the leak is missed (false negative) due to the larger distance. We provided 4 full video results in the attached video.

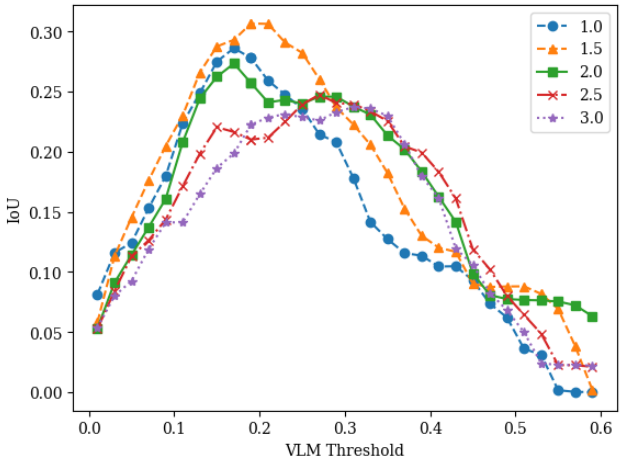

Figure 7. **Grid Search For Configuration without Background Subtraction:** We did a grid search on the enhancement factor and VLM threshold for the configuration without background subtraction. Different lines show different enhancement factors. The best performing point is when the enhancement factor is 1.5 and the VLM Threshold is 0.19. Results in this setting are values reported in Table 3.

10. Prompts Comparison

In our study on different prompts, “white steam” and “white smoke” performed the best, whereas “white plume” exhibited the worst performance. We hypothesize that the superior performance of “white steam” and “white smoke” is due to their explicit description of both the substance (smoke or steam) and its colour (white). In contrast, the poor performance of “white plume” is likely because “plume” is a relatively uncommon word.

Notably, the prompts “white gas” and “gas leak” also

performed poorly. We attribute this to the fact that, in the training data of vision-language models (VLMs), "gas" is often associated with "gas station" rather than referring solely to a gaseous substance. As a result, the model may tend to link "gas" to "gas station" or "gas stove," leading to suboptimal performance. Additionally, since gases are generally invisible in RGB images, and RGB is likely the primary modality in the training dataset, the model may struggle to associate the term "gas" with its visual characteristics in infrared imagery. This suggests that the poor performance of prompts containing "gas" is likely due to a mismatch between the term's associations in the training data and its expected visual representation in real-world scenarios.

Another notable observation is that the long prompt, "white methane leak on black background in the infrared image," achieved near-optimal performance, only slightly worse than the best-performing prompts. We hypothesize that while the VLM may not have a strong understanding of "methane," the explicit description of the black background and the infrared image modality provide sufficient context for the model to generate accurate outputs.

Algorithm 2: Background Subtraction (BGS) with Morphological Operations

Input: Video frames I_t, history length $H = 30$, scaling factor $s = 15$, threshold $T = 40$, kernel sizes for opening and closing
Output: Segmentation masks D_t

Initialize background model with H previous frames;
for *each frame* I_t **do**
  Compute background model B_t;
  Compute difference image $D_t = |I_t - B_t|$;
  Apply scaling: $D_t \leftarrow D_t \times s$;
  Threshold: $D_t \leftarrow (D_t > T)$;
  Apply morphological opening on D_t (remove salt noise);
  if *morphological closing enabled* **then**
    Apply morphological closing on D_t (merge segments);
  end if
end for

Algorithm 3: Proposed IR Gas Leak Detection Method

Input: IR video sequence $\{I_i\}_{i=1}^{N}$
Output: Segmentation masks for gas leaks

Initialize background subtraction method (e.g., MOG2);
Set $Prompt \leftarrow$ "white steam";
Set $NegativePrompt \leftarrow$ "white human, car, bird, bike";
Set VLM threshold τ_{VLM};
Set $history \leftarrow \{\}$;
for *each frame* I_i *in the sequence* **do**
  Extract background image: $I_{bg} \leftarrow \text{BGS}(I_i)$;
  Compute absolute difference: $I'_i \leftarrow |I_{bg} - I_i|$;
  Compute enhancement factor: $\alpha \leftarrow \min\left(\frac{255}{\mu_{I'_i} + \sigma_{I'_i}}, 15\right)$;
  Enhance image: $I''_i \leftarrow \text{clip}(\alpha \cdot I'_i, 0, 255)$;
  $Boxes_i \leftarrow \text{VLM}(I''_i, Prompt, NegativePrompt, \tau_{VLM})$;
  $Boxes_i \leftarrow \text{TemporalFiltering}(Boxes_i, history, \text{size}(I_i))$;
  $history = history + \{Boxes_i\}$;
  if $size(history) > 10$ **then**
    $history$.pop(0);
  Obtain masks from $Boxes_i$ using SAM 2;
  Combine all masks with OR operation to form final mask for frame i;
return Segmentation masks for sequence;