Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2026, **5**, 630

Context-aware computer vision for chemical reaction state detection

Junru Ren, [a] Abhijoy Mandal,[b] Rama El-khawaldeh,[e] Shi Xuan Leong,[f] Jason Hein, [de] Alán Aspuru-Guzik, [bcdfghijk] Lazaros Nalpantidis[a] and Kourosh Darvish*[bcd]

Real-time monitoring of laboratory experiments is essential for automating complex workflows and enhancing experimental efficiency. Accurate detection and classification of chemicals in varying forms and states support a range of techniques, including liquid–liquid extraction, distillation, and crystallization. However, challenges exist in the detection of chemical forms: some classes appear visually similar, and the classification of the forms is often context-dependent. In this study, we adapt the YOLO model into a multi-modal architecture that integrates scene images and task context for object detection. With the help of Large Language Models (LLM), the developed method facilitates reasoning about the experimental process and uses the reasoning result as the context guidance for the detection model. Experimental results show that by introducing context during training and inference, the performance of the proposed model, YOLO-text, has improved among all classes, and the model is able to make accurate predictions on visually similar areas. Compared to the baseline, our model increases 4.8% overall mAP without context given and 7% with context. The proposed framework can classify and localize substances with and without contextual suggestions, thereby enhancing the adaptability and flexibility of the detection process.

Received 6th August 2025
Accepted 28th October 2025

DOI: 10.1039/d5dd00346f

rsc.li/digitaldiscovery

1 Introduction

AI and robotics technologies provide automation solutions in self-driving labs (SDLs) to facilitate autonomous experiment execution.[1–5] Computer vision (CV), as one of the automation tools, has been applied to provide vision-based monitoring of the experiment process, sending feedback to plan future actions and make decisions.[6–8] The aim of introducing CV systems to chemical reactions in SDLs is to assist human operators in manual experiment tracking, which can be labour-intensive and time-consuming, and to standardize experiment analysis based on the macroscopic vision-captured data. In this work, we propose a real-time, context-aware chemical reaction monitoring framework.

CV has found diverse applications in chemistry and materials science, including real-time reaction monitoring, detection of physical states, materials characterization, anomaly identification, and microscopic imaging.[9] These CV systems[10,11] rely on traditional image analysis techniques (*e.g.*, edge detection, color space transformation) or/and deep learning models, particularly convolutional neural networks (CNNs).[12] Deep learning models use pixel-based inputs—typically images or video frames—and are trained for classification, detection, or segmentation tasks. However, these models operate exclusively on pixel-based data and lack any form of contextual awareness. They infer outputs by learning statistical correlations in low-level visual features—such as color, texture, edges, and spatial patterns—but do not incorporate information about the experimental protocol or materials involved. In contrast, chemists interpret visual information through contextual reasoning. Chemists' understanding of the experimental setup, the type of process underway, and the intended outcome plays a critical role in how visual scenes are classified. This missing context introduces a key limitation that CV models misclassify or inconsistently localize visually similar images that arise from fundamentally different chemical processes.

Dynamic physical processes (*e.g.*, mixing, dissolution, melting, separation, and evaporation) often generate transient,

[a]*Department of Electrical and Photonics Engineering, Technical University of Denmark, Denmark*
[b]*Department of Computer Science, University of Toronto, Toronto, ON, Canada. E-mail: kdarvish@cs.toronto.edu*
[c]*Vector Institute, Toronto, ON, Canada*
[d]*Acceleration Consortium, University of Toronto, Toronto, ON, Canada*
[e]*Department of Chemistry, University of British Columbia, Vancouver, BC, Canada*
[f]*Department of Chemistry, University of Toronto, Toronto, ON, Canada*
[g]*NVIDIA, Toronto, ON, Canada*
[h]*Department of Chemical Engineering & Applied Chemistry, University of Toronto, ON, Canada*
[i]*Department of Materials Science & Engineering, University of Toronto, ON, Canada*
[j]*Institute of Medical Science, Medical Sciences Building, Toronto, ON, Canada*
[k]*Canadian Institute for Advanced Research (CIFAR), Toronto, ON, Canada*

evolving visual cues (*e.g.*, turbidity, layering, or phase boundaries) that appear similar across different experimental setups. For example, a cloudy suspension may arise from early-stage mixing, undissolved solids, emulsified phases, or nucleating crystals, depending on the chemical experiment. Prior work with the HeinSight computer vision models[6,7,13] demonstrated real-time monitoring and control of such dynamic processes using object detection models like *R*–CNN[14] and YOLO,[15] trained on custom datasets. HeinSight organizes physical observations into a taxonomy of phase states (*e.g.*, solid, liquid, air) and tracks their interactions (*e.g.*, solid–liquid, liquid–liquid) over time to monitor process dynamics. This CV system has enabled the automation of diverse workflows such as crystallization, distillation, liquid–liquid extraction, solid–liquid mixing, solubility testing, and drug formulation across multiple platforms; from small-scale robotic systems to high-throughput experimentation and EasyMax batch reactors. However, each deployment required retraining the model on a new dataset specific to the experimental setup. These HeinSight models[6,13,14] rely solely on pixel-based analysis, which limits their contextual awareness and generalizability. As a result, they must be retrained for new contexts, especially when identical images can have different interpretations depending on the experiment. For example, in Fig. 1, the same image is labeled as two liquid layers by HeinSight 3.0 (ref. 7) (trained for liquid–liquid extraction) but as a suspended solid in liquid by HeinSight 4.0 (ref. 13) (trained for solid–liquid mixing). Both classifications are correct within their respective experimental contexts. Human annotators naturally rely on experiment type and intent when labeling such data, but this contextual information is lost during model training and inference. Additionally, Fig. 2 shows some ambiguous examples where different mixtures appear. In this work, we extend HeinSight by incorporating context into the model architecture, enabling it to differentiate experiment types and correctly classify visually similar cues based on experimental intent and hoping to resolve a general-purpose CV for chemistry.

Context-aware learning has been extensively studied as a solution to address ambiguity in tasks,[16,17] such as image classification.[18] This approach typically combines text, which provides contextual information, with images that supply visual scenes. The integration of these two inputs can occur either before or after feature extraction, referred to as early fusion and late fusion, respectively.[19] The text serves as SI, aiding in reducing the ambiguity in image features. Studies such as[20–22] have demonstrated that incorporating text inputs—providing details like geographic location, object usage, or scene descriptions—can significantly enhance performance in challenging image classification tasks. In this work, we focus on phase detection in chemical mixtures. We aim to achieve our purpose by using experiment descriptions to set a context restriction for the detection.

Vision-language models (VLMs) are a technique of multimodal systems that integrate image and text representations to perform tasks such as zero-shot classification, captioning, and open-vocabulary object detection. Models like CLIP,[23] GLIP,[24] Yolo-world[25] and Grounding-DINO[26,27] align visual and textual embeddings to retrieve or detect objects based on language prompts. However, these systems are typically trained on large-scale datasets from natural scenes, and their effectiveness in domain-specific applications—such as chemical experimentation—is limited by a domain gap in both language and visual data. Moreover, most VLMs use text as a search query to localize corresponding visual elements, which differs from the context-guided classification required in chemistry. Our approach builds on VLM principles but reframes the role of text. In YOLO-text, textual input (*e.g.*, experimental protocols) is used as a context signal rather than a retrieval query. By attaching a context-aware learning block to the YOLO detection head, our model fuses textual and visual features during training and inference. This enables the model to adjust its predictions based on the experiment's intent, allowing more accurate classification of visually similar inputs across different workflows.

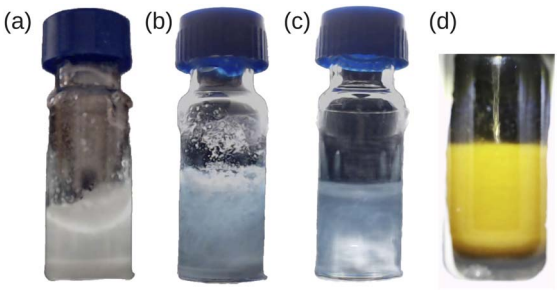


Fig. 2 The detection of chemicals can often be ambiguous due to the similar visual features shared by different forms. Moreover, classification may rely on context, which is influenced by the type and objective of the experiment. Vial (a) illustrates the process of solid particles settling at the bottom of the vial, whereas vial (b) depicts the aggregation of solids within a heterogeneous liquid. Vial (c) demonstrates a heterogeneous liquid with a non-uniform distribution, and vial (d) illustrates detection from a commonly used chemistry reactor set, characterized by a non-transparent yet uniformly distributed homogeneous liquid. Although the liquids in vials (a) and (b) are visibly heterogeneous, the detection of solids remains ambiguous. Similarly, the liquid classes in vials (c) and (d) exhibit overlapping visual characteristics, complicating their differentiation.

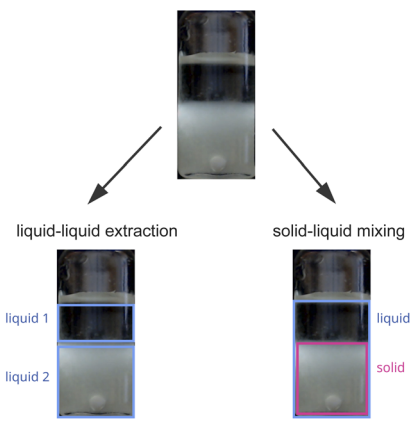


Fig. 1 An example of a single image can be labeled by different classes under different chemistry contexts.

In this way, YOLO-text adapts the strengths of VLMs to chemical CV tasks while addressing the limitations of open-vocabulary approaches in scientific domains.

In this work, we present YOLO-text, a real-time, context-aware vision-language model for detecting phase states in dynamic chemical experiments. By integrating a pretrained large language model (LLM), YOLO-text incorporates textual input (*e.g.*, experimental protocols) to guide visual interpretation, particularly in cases where identical visual cues lead to different classification outcomes depending on the experimental context. YOLO-text model includes five relevant physical phase classes commonly encountered in chemical workflows: gaseous headspace ("empty" or "residue"), liquid ("homogeneous" clear or "heterogeneous" cloudy) and "solid". To enable context-aware detection, we introduce a lightweight context-aware learning block that connects to the YOLO backbone[28] and optionally accepts text prompts as input. This block fuses visual features extracted by the vision model with textual cues before passing them to the detection head, forming a unified image-text multimodal fusion model. This design supports real-time inference while allowing flexible use of context: the model can be trained and deployed with or without textual input. YOLO-text employs a two-stage training strategy: first, it aligns image and text representations into a shared embedding space; then it learns to detect phase states under different context conditions. We show that introducing context dramatically improves detection accuracy, especially for underrepresented classes. For instance, mean average precision (mAP) for the solid class improves from 27.3 to 54.9. Overall, YOLO-text outperforms the standard YOLO model across all classes, with total mAP increasing from 75.9 to 82.9 with context, and 80.7 without it. We demonstrate YOLO-text on four case studies each involving visually similar images that require different classifications depending on the experiment type. YOLO-text lays the foundation for context-aware classification in chemistry, enabling the same image to be interpreted differently based on experimental intent and marks a step toward building general-purpose computer vision systems for chemical research.

2 Experimental

The following sections describe the design of the model architecture and the dataset. The Dataset Section outlines the challenges encountered in constructing the training dataset and explains the selection and annotation of five classes from visual data obtained through various chemistry experimental techniques. The Method Section presents the components proposed and implemented within the model architecture.

2.1 Dataset

Data collection is conducted from real chemical experiments in laboratory settings. The dataset includes experiments from various chemical techniques to ensure the diversity of each class. However, certain classes, such as solid and residue, are inherently less common. To address the challenges of small and imbalanced data, the dataset annotation is designed to use a minimal number of classes that can effectively represent the material phases typically observed in experiments. Therefore, the dataset is annotated using five classes: empty layer, residue, solid, homogeneous liquid, and heterogeneous liquid. The definitions for each class are as follows:

- Empty: air layer in the vessel.
- Residue: solid particles sticking on the vessel window.
- Solid: big solid chunks suspended in liquid and/or solid sediments.
- Homogeneous liquid: transparent and liquid with uniform composition.
- Heterogeneous liquid: not uniformly distributed liquid.

To give an insight into the dataset, we show several images from the dataset with their annotations in Fig. 3. It is important to note that, in some cases, different classes can be inclusive. For example, a homogeneous liquid may contain a solid, leading to visual overlap between classes.

A total of 17 videos, along with the data used in HeinSight 3.0,[7] were recorded from various experiments or different stages of an extended reaction. The videos encompass four commonly used techniques in chemistry experiments: liquid–liquid extraction (LLE), solid–liquid mixing, crystallization, and dissolution, as well as scenarios involving empty vessels with a static or operating stirring bar. Specifically, 828 images and 3 videos were derived from the LLE process, 6 videos were recorded during the solid–liquid mixing process, 2 videos captured the crystallization process, and 2 videos documented the dissolution process. Fig. 4a illustrates the portion of classes from the training set contributed by different chemistry techniques.

Image data is extracted from recorded videos, where we first separate 10 videos into the training set and 7 videos into the validation and test set, and then extract image data with a certain frame rate. In the end, there are 2841 images in the training set, and 280 images in the test set. Fig. 4b shows the number of samples of each class in the training and test sets.

Fig. 2 is a glimpse of some difficult image examples in our data set, where (a), (b), and (c) were directly collected from vials and (d) was collected from EZ-Max experimental equipment. The data shown in Fig. 2 were recorded from different experiments showing varied chemical layers with similar visual cues. For example, (c) and (d) are non-transparent liquids annotated as heterogeneous and homogeneous liquids, respectively. The visual ambiguity

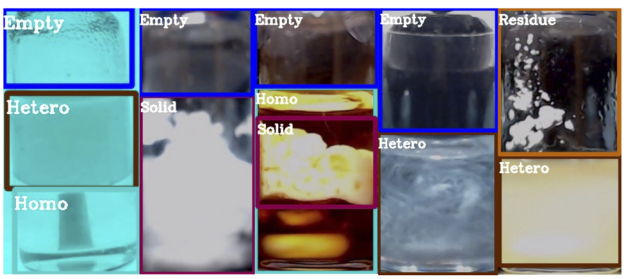
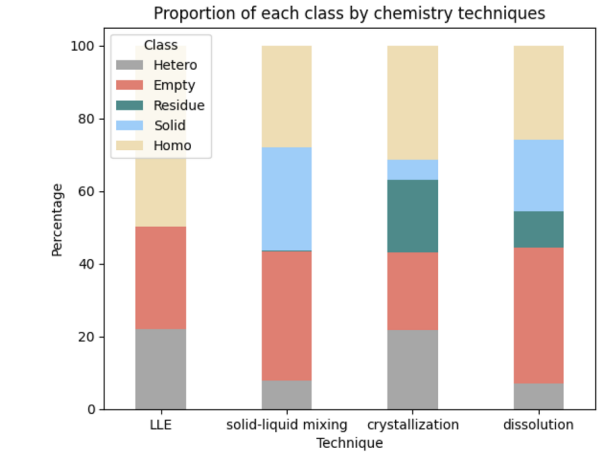


Fig. 3 Some examples of images and their annotations from the dataset. The background of each image was cropped during training.



(a)

(b)

Fig. 4 Two bar plots to illustrate the data distribution (a) is an illustration of the portion of data in the training dataset coming from different chemistry experiment processes. (b) shows data distribution for each class in the training and test sets. In (b), light red bars refer to the number of samples in the training set, while grey bars refer to the samples in the test set. Overall, the data is split between the training and test sets at a ratio of approximately 10 : 1

might confuse the model during training, and make it provide inaccurate predictions under different experimental contexts.

2.2 Method

In this section, we present a chemical reaction monitoring model, as illustrated in Fig. 5. A pretrained large language model (LLM) is employed to process the ongoing experiment protocol and infer the expected classes. The expected detections generated by the LLM are then passed to YOLO-text, which subsequently predicts states based on the fusion of visual and contextual information. To mitigate the uncertainty in LLM reasoning, YOLO-text is trained to adapt to varying levels of contextual input, including complete, incomplete, and absent class information.

We will start by introducing the architecture of YOLO-text. As shown in Fig. 6, the model takes images and offline words (context) as input, making predictions based on vision features and context. Then, a description of how we conduct prompt tuning on the pretrained LLM to generate output that can be taken as input by YOLO-text is provided.

2.2.1 Image features and word embeddings

2.2.1.1 Image feature extraction. The proposed model, YOLO-text, is built upon YOLOv8. It retains the original vision feature extraction process established by YOLO,[29] utilizing a Darknet backbone as the image encoder to extract multi-scale features. These features are subsequently processed by the Feature Pyramid Network (FPN)[30] to enhance feature representation.

2.2.1.2 Word embeddings. Each word input to the text encoder is initially transformed into an embedding. At the start of training, these word embeddings are initialized using pre-trained global word vectors.[31] The embeddings are stored in the PyTorch Embedding layer and updated throughout the training process. To handle varying input word counts, we define a fixed maximum number m of acceptable inputs as the dimension of the text embedding matrix. The input word indices $W = \{w_1, w_2, \ldots, w_n\}$ must satisfy $n < m$. Each input word w_i is mapped to an embedding vector e_i and stored in the embedding matrix $E \in \mathbb{R}^{|V| \times d}$, where $|V|$ is the vocabulary size and d is the embedding dimension:

$$e_i = E[w_i], \ \forall i \in \{1,2,\ldots,n\}$$

The extracted embeddings are arranged into a fixed-dimension tensor $T \in \mathbb{R}^{m \times d}$, with padding as needed:

$$T = \begin{bmatrix} \\ e_1 \\ e_2 \\ \vdots \\ e_n \\ e_{\text{no-prompt}} \\ \vdots \\ e_{\text{no-prompt}} \\ \\ \end{bmatrix}$$

2.2.2 Context-aware learning block.

The context-aware learning block is positioned before the detection heads, where it integrates textual and multi-scale image features, enabling the fusion of these two modalities before passing them to the detection heads. Within this module, the text features are initially projected into the same latent space as the image features and then concatenated with the image features to incorporate visual information. Both types of features are subsequently processed through a self-attention module to enhance their individual representations. The context-aware learning block employs two types of fusion processes: weighted embedding fusion and joint embedding fusion. The weighted embedding fusion process aligns the two feature types by calculating their similarity using scaled dot-product cross-attention. The resulting weighted embeddings are then

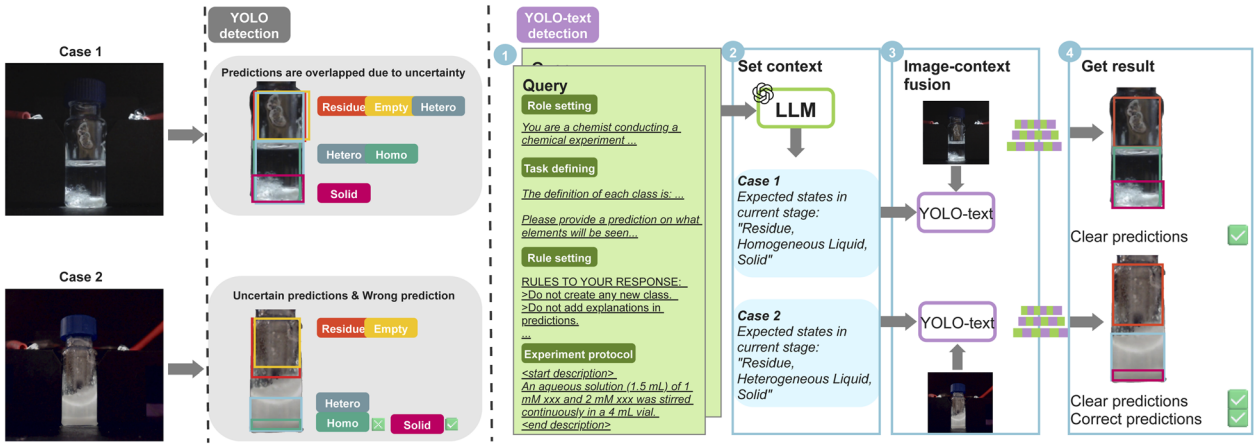


Fig. 5 Diagram illustrating the proposed framework for chemical reaction detection. The figure on the left shows two examples of failed detection results from a vision-only detection model. There is no Non-Maximum Suppression (NMS) used in detection, as it can mistakenly remove the overlapped objects, such as a solid in a homogeneous liquid. The right of the figure shows the working procedure of our proposed method. According to our method, a pre-trained LLM is used to reason the reaction phases from the provided experiment protocols. The LLM reasoning result is used to set the context for YOLO-text, which outputs the final detection results on visual frames. Image and context are fused in YOLO-text, where the predictions are adjusted based on the set context. Notably, while the input context aids the detection process, it does not solely determine the final predictions. This ensures that the model can still generate predictions even in cases where context input is unavailable or incomplete from the LLM.

merged through summation. Empirical analysis indicates that this approach improves sensitivity to detailed pixel variations but may lead to an increase in false positives. The joint embedding fusion process, in contrast, applies a concatenation operation to combine the features globally, providing a more holistic fusion strategy.

2.2.2.1 Decision making. The decision-making function is inserted in the context-aware learning block to weigh the embeddings passing to the detection heads. It will determine the portions of the features from the two fusions and forward them to the detection heads for final predictions. Two decision-making approaches have been implemented: a decision-making function and a concatenation operation. The following sections provide a detailed explanation of each method.

F_1 and F_2 are given as the two fused embeddings from the previous calculations, where F_1 refers to the weighted embedding fusion and F_2 refers to the joint embedding fusion.

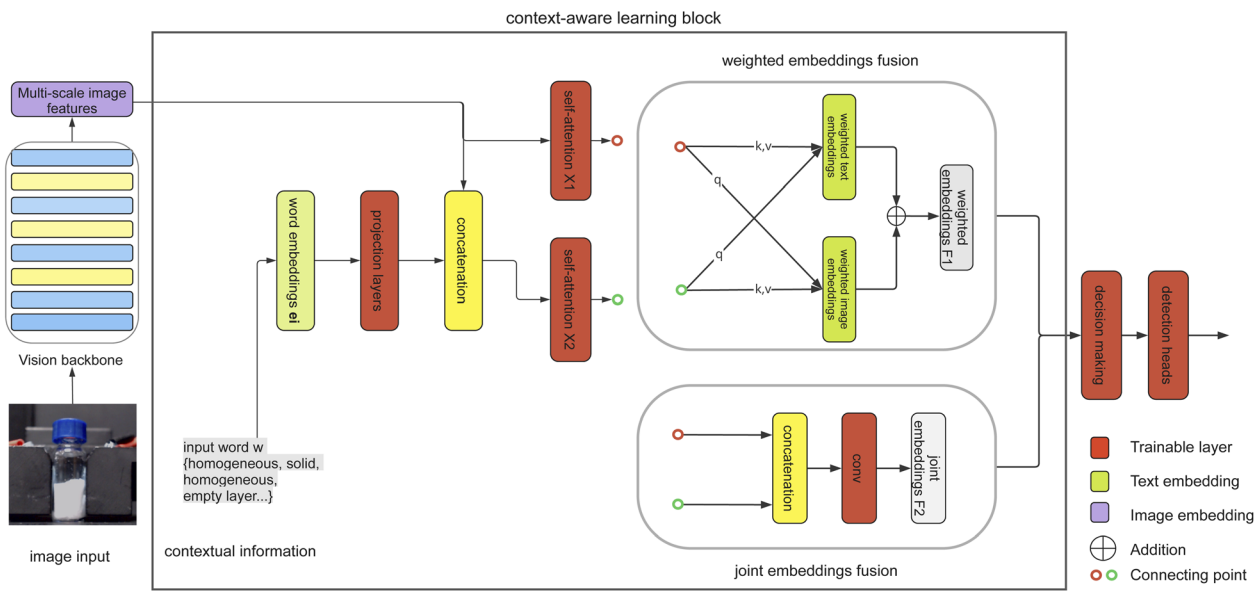


Fig. 6 Architecture of the proposed context-aware learning block. The original YOLO backbone is used to extract vision features. The context-aware learning block takes multi-scale image features and context information as input. The two modalities are fused as weighted embeddings and joint embeddings, and then sent to the decision-making block. Lastly, the decision-making block selects the fused features and passes the features to detection heads.

The first method, the decision-making function, comes from the fusion mechanism proposed in,[20] where we combine the two features by eqn (1).

$$F = w \otimes F_1 + (1 - w) \otimes F_2, \quad (1)$$

w is the weight of different features, which is calculated by the eqn (2). $\otimes$ represents the element-wise product. $w = \sigma(W_1 X_1 + W_2 X_2)$ (2)

X_1 and X_2, respectively, represent the pure image features and word embeddings output from the self-attention modules; W_1 and W_2 are two linear projection layers; σ is the sigmoid function to project the coefficients to the range between 0 and 1.

The second decision-making approach is to employ a convolutional layer to learn how to select the features from two fusions.

$$F_c = C(F_1, F_2), F = \mathrm{Conv}(F_c) \quad (3)$$

As shown in eqn (3), the two features are initially combined in the concatenation layer C to form the unified feature representation F_c. The resulting F_c is then passed through a convolutional layer to be projected into the appropriate dimensional space.

2.2.3 LLM prompting. Prompt engineering is employed to guide and refine the outputs of the LLM. The prompting process begins by assigning the LLM a specific role and task: to perform general predictions by reasoning about the expected classes that may emerge during the experimental stage, based on the provided experimental protocol. To ensure the LLM's outputs are compatible with the proposed model, the prompts also include explicit prediction rules that constrain and structure the responses.

2.2.4 Mixup. The limited size of the training set increases the risk of overfitting during model training. To ensure that the input context effectively guides the prediction, we introduce visual ambiguity into the dataset. Specifically, the mixup data augmentation technique[32] is employed to achieve this goal. This technique combines two batches of images at the pixel level, with a randomly generated value λ determining the mixing intensity for each operation. The data $\bar{x}$ after mixing can be represented by the eqn (4), where x_i and x_j are two image batches.

$$\bar{x} = \lambda x_i + (1 - \lambda) x_j \quad (4)$$

$$\bar{y} = \lambda y_i + (1 - \lambda) y_j \quad (5)$$

$$\mathscr{L} = \lambda \mathscr{L}(P, T_A) + (1 - \lambda) \mathscr{L}(P, T_B) \quad (6)$$

In addition, the labels $\bar{y}$ will be mixed by the same mechanism as eqn (5).

Table 1 includes several extensions of the mixed methods. The image data are mixed by λ, but the labels are selected as either the batch with higher λ or the union of two batches. Text sent to context-aware uses the same strategy as the label. Loss function uses either the original formula or is weighted by λ as eqn (6), where P represents prediction result and T is the ground truth, $\mathscr{L}(P, T_A)$ is the original loss of batch A and $\mathscr{L}(P, T_B)$ is loss for batch B. Additionally, other data augmentation techniques are applied during training to further diversify the dataset; these will be discussed in detail in the next section.

Table 1 Mixup augmentation setting

Images	Labels	Text	Loss
Mixed	High λ	High λ	Original
Mixed	Union	Union	Original
Mixed	Union	Union	Eqn (6)
Mixed	Union	Union, λ	Eqn (6)

3 Results and discussion

3.1 Experiments and results

3.1.1 Training steps. A two-step training strategy is employed to train YOLO-text. Before training, YOLO-text loads the pre-trained YOLO weights into the vision backbone.

3.1.1.1 Context-aware training. In the first step, the entire model is trained on the custom dataset, with all ground truth class names provided as prompts during training. During evaluation, the class names are also input as prompts for the test data. The expected output is the class names and locations of the bounding boxes. This step teaches the model the semantic meanings of each class name and enables the model to provide predictions with hints from the prompted class.

3.1.1.2 Vision-aware training. In the second step, the model is fine-tuned either without class prompts or with a random subset of class prompts. This step trains the model to adapt to varying prompting scenarios, ensuring it can produce robust predictions even when prompts are incomplete or absent.

3.1.2 Evaluation metrics. We adopt Precision (P), mean Average Precision (mAP) and recall (R) as the metrics to evaluate the model performance in the object detection task for each class. Especially when the class area is difficult to detect, for example, the solid class is always shown in various shapes and colors, the recall rate is an essential factor in measuring the model's reliability on those objects.

3.1.3 Implementation details. We train models using a linearly decreasing learning rate, with an initial value of 0.05 for text encoder and visual encoder training and an initial value of 0.005 for decision-making layers training. The decreasing rate is set as 0.001. Stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and a weight decay of 0.005 is implemented. In the context-aware learning block, we add dropout layers with a ratio of 0.5 to prevent overfitting. The rest of the hyperparameters are set as the system default, tunable for potentially better results. As our training dataset is small, we load the pretrained weights on the COCO dataset to the visual backbone and the detection heads. Data augmentation is also implemented by the Albumentations library.[33] Specifically, we use data affine, flipping, data blur, noise, and color jitter augmentation with a probability of 0.5. In the first training step, we introduce data mixup to increase the ambiguity in the

Table 2 Experimental results on our dataset. All grounded labels are prompted during our model training. The best performances are marked with bold text. In the table, *P* refers to Precision, *R* refers to Recall, and mAP refers to mean average precision. Faster *R*–CNN uses ResNet50 as backbone. D1 refers to the weighted function for decision-making. YOLO-text-wAdd uses output from weighted embedding fusion. YOLO-text-joint uses the output from joint embedding fusion. YOLO-text-D1 takes both weighted and joint fused embeddings and uses a decision-making approach described in the eqn (1).YOLO-text uses the decision-making approach described in the eqn (3)

	Overall				Hetero		Empty		Residue		Solid		Homo	
Methods	mAP 50 : 95	mAP50	*P*	*R*	mAP 50 : 95	*R*	mAP 50 : 95	*R*	mAP 50 : 95	*R*	mAP 50 : 95	*R*	mAP 50 : 95	*R*
Faster *R*–CNN	37.4	60.1	66.8	51.5	59.0	63.9	67.2	82.3	29.1	57.0	10.4	17.0	12.6	16.8
YOLOv8	75.9	84.3	88.8	81.7	96.1	96.4	90.8	94.6	83.8	94.1	27.3	37.0	76.6	86.4
YOLO-world	81.2	86.8	94.9	80.0	94.3	89.0	90.0	95.0	**92.7**	95.0	39.8	34.0	89.1	88.6
YOLO-text-wAdd	75.7	87.0	97.1	84.9	92.9	**99.9**	93.7	99.2	84.3	**99.9**	28.7	33.1	78.9	92.0
YOLO-text-joint	79.1	88.7	98.0	85.3	93.2	**99.9**	**95.5**	**99.9**	89.4	**99.9**	32.6	36.4	84.9	90.0
YOLO-text-D1	84.0	91.2	**98.0**	85.1	97.2	99.3	95.1	98.4	90.7	**99.9**	45.5	35.2	**91.7**	**92.7**
YOLO-text	**85.0**	**97.6**	96.9	**96.7**	**97.8**	**99.9**	93.3	**99.9**	89.4	**99.9**	**54.9**	**93.8**	89.9	89.8

training data. Note that we disable the mosaic augmentation during training, which is set to true by default in the YOLO trainer. This is because we would like to keep the relative positions between areas. Additionally, agnostic max non-suppression is also disabled, as there are existing situations where one class is visually overlapping.

3.1.4 Model performance in image detection

3.1.4.1 Evaluation result after context-aware training step. Model performance is first evaluated with images in the test set. Since YOLO-text is developed on the YOLOv8 backbone, and YOLO-world is regarded as a state-of-the-art YOLO-based VLM, our model is compared against YOLOv8 and YOLO-world as baselines. The two baselines were trained with the same hyperparameter settings as YOLO-text but with the learning rate adjusted to prevent underfitting and overfitting. Additionally, the input text prompt is sent to YOLO-world by default. Table 2 shows the models' performance in each class. YOLO-text-wAdd only uses output from weighted embedding fusion and directly passes the output to the detection head, while YOLO-text-joint uses the output from joint embedding fusion. YOLO-text-D1 takes both weighted and joint fused embeddings and uses a decision-making approach described in the eqn (1). YOLO-text, which uses the decision-making approach defined in eqn (3), is the selected model.

We notice that for the classes heterogeneous liquid, empty and residue, all architectures perform well in terms of mAP and recall. In contrast, YOLOv8 shows relatively low mAP and recall in the solid and homogeneous liquid classes. This is because the number of solid samples in the training set is relatively small compared to others (only about 700 samples), and both solid and homogeneous liquids in the test set contain different visual features compared to the training set. We can see that using prompted context in YOLO-text can help the model locate the corresponding features. As a result, the proposed model can maintain or slightly improve detection performance for the heterogeneous liquid, empty, and residue classes (ranging from 1% to 6%) while achieving significant improvements in detecting solids and homogeneous liquids, with increases of approximately 10% to 25%.

Additionally, the performance of YOLO-world on the test set is evaluated. Overall, YOLO-world demonstrates strong performance, achieving an mAP of 81.2. Notably, the mAPs for the solid and homogeneous liquid classes are higher compared to YOLO; however, the recall rates for these two classes are significantly lower than those of other classes. This discrepancy is likely due to YOLO-world's training objective as a VLM, which focuses on contrastively matching distinct pairs of text and images rather than using text as contextual hints to enhance

Table 3 Experimental results of the model training and testing with different prompting strategies. The best performances are marked with bold text. D1 refers to the weighted function for decision-making, and D2 refers to the concatenation with a convolutional layer for decision-making

	Prompting during evaluation											
	All labels				Random labels				No prompts			
Prompt strategies	mAP50 : 95	mAP50	*P*	*R*	mAP50 : 95	mAP50	*P*	*R*	mAP50 : 95	mAP50	*P*	*R*
All labels (D1)	84.0	92.1	**98.0**	85.1	75.0	86.1	81.1	83.2	28.5	34.1	38.0	27.6
All labels (D2)	**85.0**	**97.6**	96.9	96.7	74.5	89.9	87.4	80.8	24.7	42.9	29.1	50.8
Random labels (D1)	81.7	90.3	95.1	82.6	81.3	89.7	89.0	79.7	77.7	85.4	68.7	72.3
Random labels (D2)	68.2	80.1	93.4	80.9	70.1	81.4	92.9	80.8	68.2	81.1	71.8	79.0
Finetune w/o prompting (D1)	81.1	87.4	95.2	84.2	80.6	87.9	**94.2**	83.6	77.5	84.3	85.7	73.9
Finetune w/o prompting (D2)	82.9	95.6	92.2	**91.3**	82.3	92.5	90.0	87.5	**80.7**	**89.8**	**85.9**	85.8
Finetune with single labels (D1)	80.9	91.7	92.2	86.7	79.3	88.8	90.9	82.1	73.9	80.2	51.0	85.9
Finetune with single labels (D2)	75.3	88.5	88.3	84.0	71.7	83.5	84.1	75.0	65.1	71.3	41.8	82.6
Fintune with random labels (D1)	83.3	94.1	91.7	**91.3**	**82.1**	**91.2**	93.5	**84.3**	68.6	76.1	43.4	**86.3**
Fintune with random labels (D2)	79.5	90.2	92.2	84.9	80.3	90.8	87.6	86.9	70.3	77.5	44.6	86.0

classification and localization tasks. However, further research and verification are required to confirm this hypothesis. Although YOLO-World achieves a high mean Average Precision (mAP) in the chemical detection task, it does not meet our requirements. Specifically, our approach aims to use text prompts to (i) correct incorrect detections, (ii) recover missed detections, and (iii) preserve accurate detections. As YOLO-World aims to support open-vocabulary detection, it performs detection based on a specific text prompt, limiting its flexibility in addressing these objectives.

3.1.4.2 Evaluation result after vision-aware training step. While providing all ground-truth labels as prompts can validate the model's capabilities, this approach alone does not fully leverage its potential. The model is expected to generate accurate predictions without prompts and dynamically adjust its outputs based on input prompts, all while preserving correct detections. Table 3 demonstrates the robustness of YOLO-text under various prompting strategies, including no prompts, randomly selected labels, and all labels. It highlights how different prompting strategies used during training influence the model's performance under corresponding inference conditions. Following the context-aware training, the model achieves the highest mAP when provided with all category names, with performance declining under random or absent prompts. To address this, vision-aware training is introduced, where the model is fine-tuned using diverse prompting conditions to better align with the task requirements. Compared to all results, the D2 method, which fine-tunes all model layers when no valid prompt is available, is selected as the most effective. It achieves the highest mAP of 80.7 in the no-prompts scenario and maintains strong performance with an mAP of 82.3 under random prompts.

Fig. 7 provides examples illustrating the performance of YOLO and YOLO-text on the test set. The figure consists of two rows of images. The first row provides the prediction results from YOLO that are wrongly detected and misdetected. The first two images from the left show that the YOLO detects the stirring bar at the bottom as the solid class, which has been removed from our model's prediction. The remaining images in the first row show the wrong detections on ambiguous areas, which are corrected by our model with the contexts provided during training. The second row of images includes examples marked with red rectangles, indicating cases where category names were prompted during inference to refine classification and detection. We notice that even though our model has removed the wrong predictions in the last image, it still fails to detect the individual liquid layers. This limitation highlights an area for further improvement in future research.

3.1.5 Model performance in video detection. In this section, we present two case studies to demonstrate how the model monitors the experimental process using video data and to compare its performance under different contextual settings.

3.1.5.1 Sedimentation process. This case demonstrates a sedimentation process captured from a chemical lab instrument called EasyMax. The experiment begins with a homogeneous solvent with solid setteled at the bottom of the vessel.

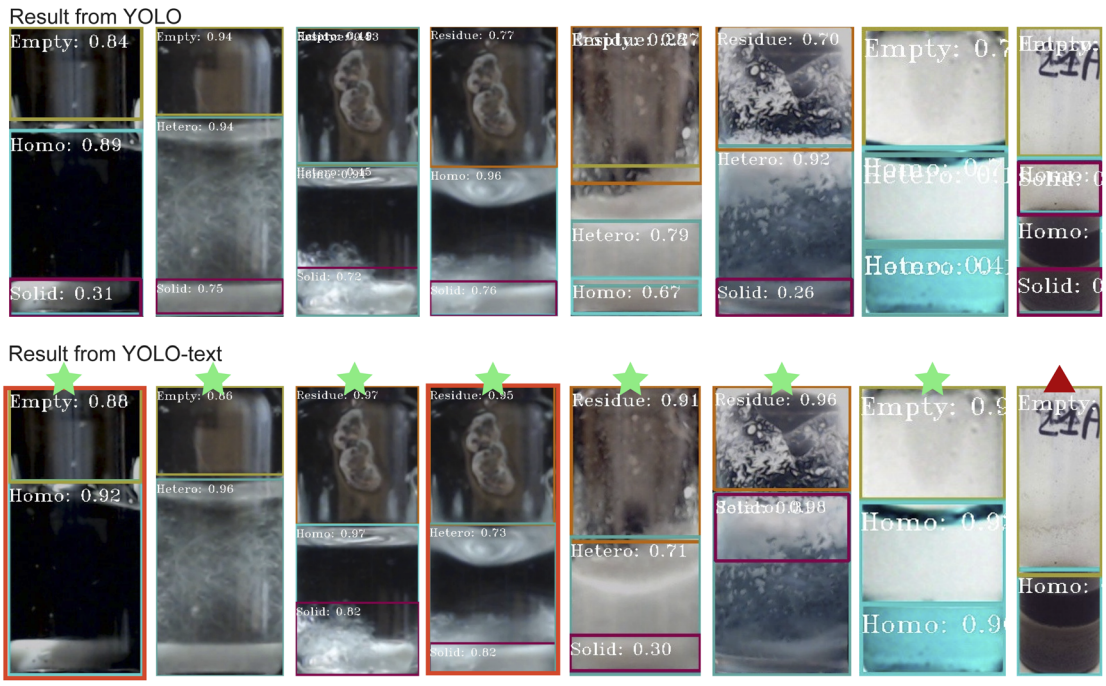


Fig. 7 Visualization examples illustrate how the proposed model corrects its predictions on ambiguous areas. The images in the first row are the wrong predictions from YOLO-v8. The images in the second row marked with red rectangles show the result when the context is provided, while the images without red edges are from the proposed model but without the context provided. The green star represents the correct prediction based on annotations, while the red triangle indicates the wrong prediction. In the red-triangle-marked figure, the first image to the right has two separate liquid layers at the bottom of the vial; however, the proposed model can only detect one. The first image to the right is taken from ,[34] and the second image to the right is taken from.[35]

When the stirring bar is activated, the clear homogeneous liquid and solid are mixed together to form a heterogeneous liquid. When the stirring bar stops, solid particles within the heterogeneous solvent gradually settle at the bottom of the vessel. Fig. 8 illustrates the progression of case A, aligning visual scenes with a corresponding plot at the bottom that shows the model's detection results over time. The *y*-axis of the plot represents the height of the predicted bounding boxes, with different colors indicating different predicted classes. At the start of the experiment (0–4 seconds), the prompt "homo, residue" is provided to guide the model's predictions. However, the model mistakenly classifies the stirring bar as solid during this stage. In the middle of the experiment, the prompt is updated to "hetero, residue," and from 10 seconds onward, "solid" is added to enhance the model's prediction accuracy. Fig. 10 compares the results from YOLO and our proposed model under different prompt settings. In Fig. 10a, YOLO's output shows that the solvent is correctly identified in some frames during the initial stage. However, in subsequent stages, while the model predicts the appropriate classes, the bounding box locations remain unstable. Fig. 10b displays the predictions from our proposed model without any prompts. In this case, the model tends to predict all potential classes in ambiguous regions and incorrectly classifies residue as empty during the latter half of the experiment. This issue is resolved in Fig. 10c, where the prompt "residue" is provided, leading to corrected predictions. Finally, Fig. 10d evaluates the model's behavior when the prompt "hetero" is consistently applied throughout the experiment. This setting tests the model's ability to adapt its predictions based on context and resolve overlapping detections effectively.

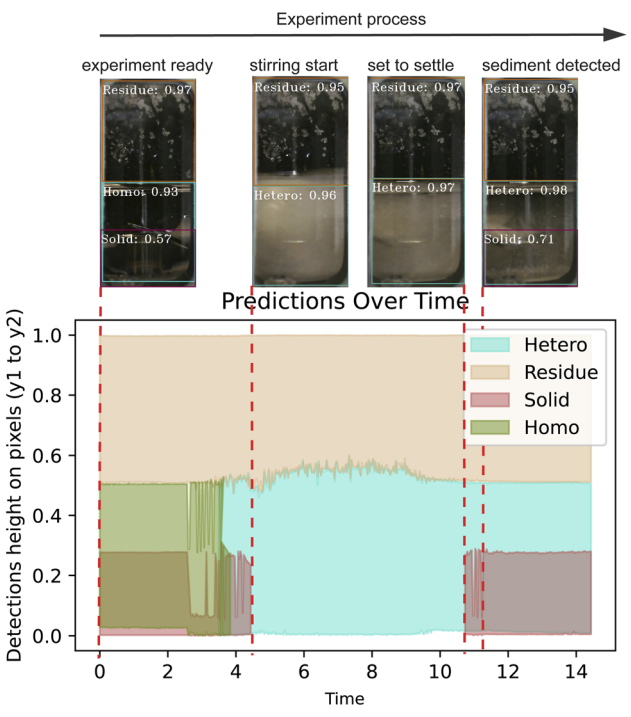


Fig. 8 Sedimentation case: a demonstration of the reaction process described in case A. The images match the detection plot from YOLO-text. The *X*-axis shows the time in seconds, while the *Y*-axis is the normalized height of the bounding boxes. The colors represent different detected classes.

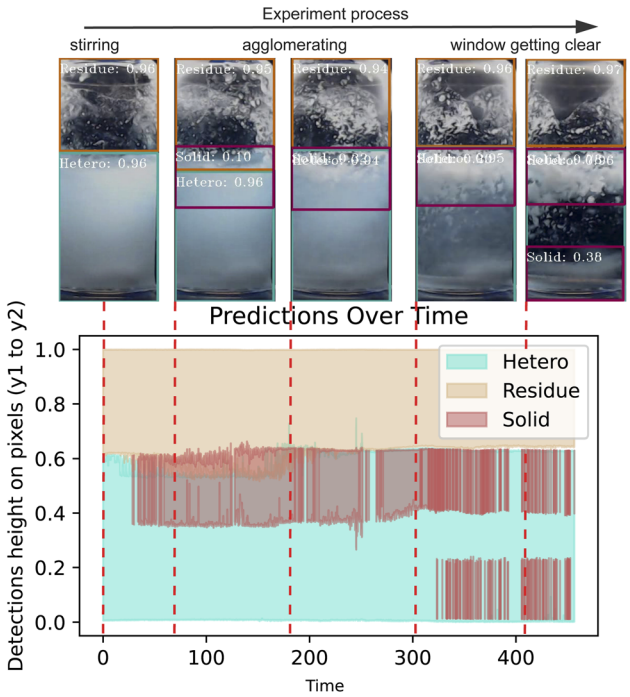
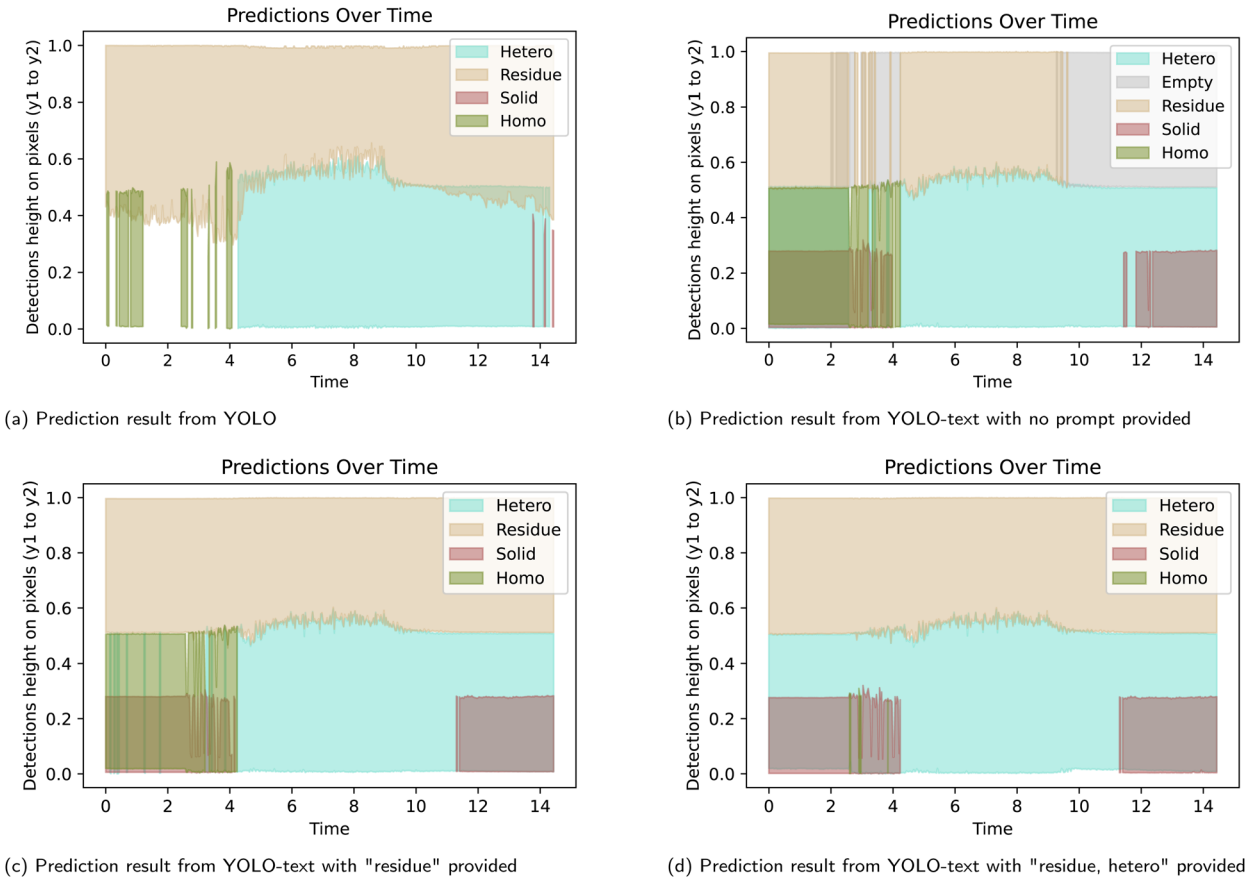


Fig. 9 Agglomeration process: a demonstration of the reaction process described in case B. The images match the detection plot from YOLO-text. The *X*-axis shows the time in seconds, while the *Y*-axis is the normalized height of the bounding boxes. The colors represent different detected classes.

3.1.5.2 Agglomeration process. During the process, the system begins with a heterogeneous mixture of small solid particles suspended in a liquid. The particles begin to agglomerate and form a layer at the top of the solution with stirring, leaving behind a clear homogeneous liquid. Fig. 9 illustrates the process. Through the monitoring, "residue, hetero, solid" are prompted to the model. The model is able to detect the agglomerate when it starts to form. However, after the liquid turns transparent, the model again detects the stirring bar as another solid object. Fig. 11 compares the prediction results from YOLO and our model. YOLO provides two predictions on the liquid area in the first 300 seconds and gives wrong predictions on the solid and semi-transparent solutions. In our model, the overlapping prediction can be fixed by context setting, which also leads the model to detect the generated solid.

3.2 Discussion

3.2.1 Data imbalance. Data quality is important in every computer vision task, especially when custom datasets need to be developed. The number and diversity of the training data, as well as the relations between the training and the test set, will significantly influence the model's performance on specific tasks. However, the research field of computer vision for chemical lab automation lacks large-scale public datasets capturing visual changes during real chemistry experiments. The lack of sufficient data is our biggest challenge. In Section



(a) Prediction result from YOLO

(b) Prediction result from YOLO-text with no prompt provided

(c) Prediction result from YOLO-text with "residue" provided

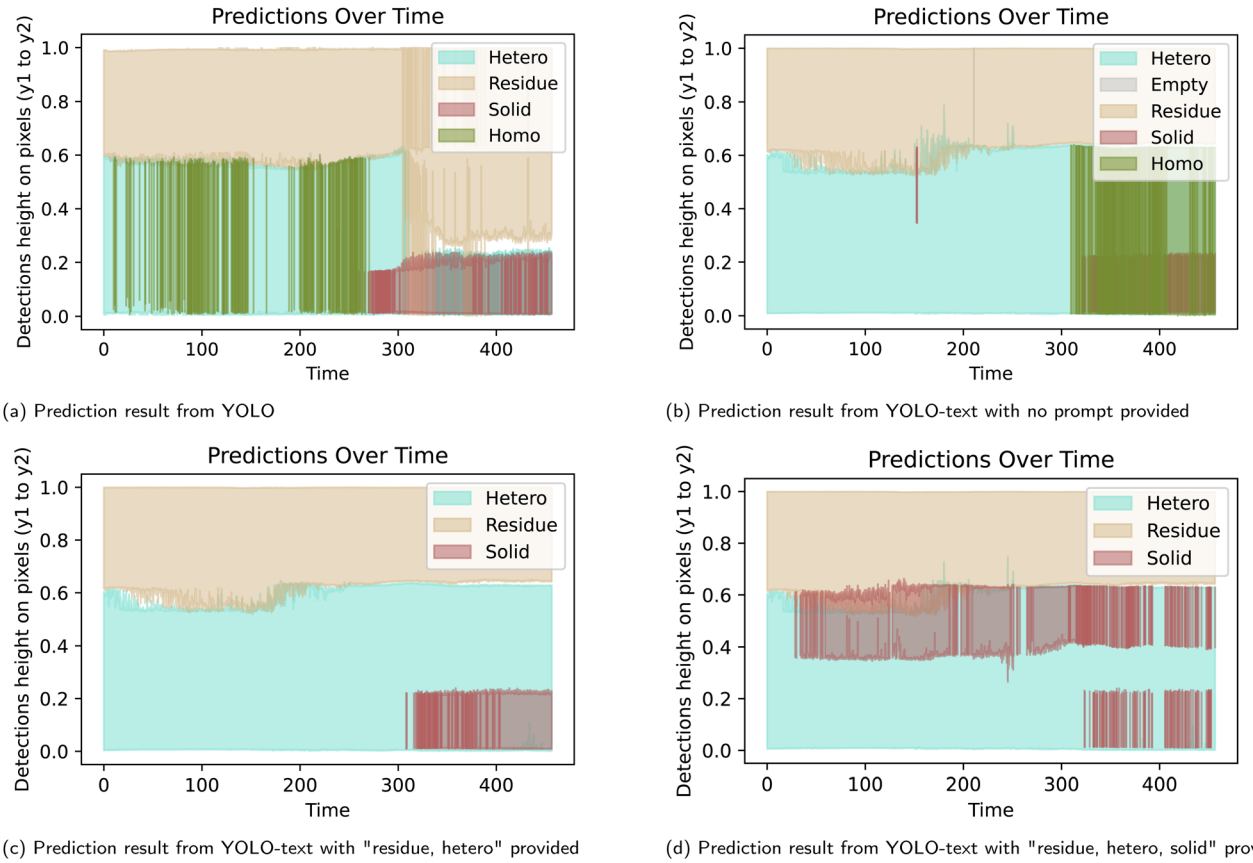
(d) Prediction result from YOLO-text with "residue, hetero" provided

Fig. 10 Comparison of sedimentation process detection results using YOLO and YOLO-text under different prompting conditions. (a) presents the detection results from YOLO, which exhibits incomplete predictions at both the beginning and end of the experiment. (b) shows the output from YOLO-text without prompts, demonstrating high sensitivity to scene changes. In (c) and (d), YOLO-text is prompted with contextual cues, leading to altered predictions based on the provided input. When given the prompt "residue," the model shows increased confidence in classifying the upper area as residue. Similarly, when prompted with "hetero," it removes the "homo" class; however, this results in an incorrect prediction relative to the annotation. The actual experiment process is shown in Fig. 8.

2.1, we describe how to efficiently separate data into training and test sets. We first extract frames from all experiment videos that we have and then randomly separate frames into training and test sets. The vanilla YOLO model can provide very high mAP in each class under this data structure due to the high similarity of the two sets. However, the model performance will sharply decrease when testing on new experiments. Therefore, we first separate the videos into training and test sets, and then extract image frames to ensure the model's reliability when encountering previously unseen experimental scenarios during inference. However, it is undeniable that the similarity between the frames from the same video still exists in each set, which could cause a dramatic drop in precision if the model fails to detect in one specific case; this is what happened to the "solid" detection in YOLO. The solid substance is naturally less common in chemical experiments than the other categories, and when we organize the dataset, the aim is to separate different experiments, which also causes a decrease in the diversity of the solid.

The data imbalance came from the inherent characteristics of chemical experiments, which can lead to biased prediction performance. To address this issue, we adopted a combination of data augmentation and bias-reducing loss functions in model training. Despite the application of various data augmentation techniques, their effectiveness in enhancing the diversity of solid appearances remains limited. This limitation motivates the incorporation of contextual information into the model to better support the detection of previously unseen visual features. In addition, the dataset includes experiments captured under different experimental setups, which naturally contributes to increased training diversity and further aids generalization.

3.2.2 Leveraging text prompts to refine visual detection. VLMs are typically designed to perform open-vocabulary detection by using textual descriptions to localize prompt-relevant regions in an image. In our case, the text helps to fix the prediction result in the related area. The model keeps its ability to make predictions based on visual cues when no contextual information is provided. The LLM introduces contextual reasoning capabilities that allow it to interpret experimental descriptions more flexibly and handle complex procedures that may not be captured by direct keyword



(a) Prediction result from YOLO

(b) Prediction result from YOLO-text with no prompt provided

(c) Prediction result from YOLO-text with "residue, hetero" provided

(d) Prediction result from YOLO-text with "residue, hetero, solid" provided

Fig. 11 Comparison of agglomeration detection results using YOLO and YOLO-text under different prompting strategies. (a) shows the detection results from YOLO, which exhibits unstable predictions throughout the experiment. (b) presents the YOLO-text output without prompts, showing improved performance by reducing incorrect detections of homo classes. In (c) and (d), prompting with "residue" and "hetero" stabilizes the boundary between the upper and lower regions. Additionally, when prompted with "solid," the model begins to interpret the agglomerates as solid materials over the course of the experiment. The actual experiment process is shown in Fig. 9.

matching. In addition, integrating an LLM makes the framework more user-friendly. It shows great potential of integrating LLM to assist visual detection, especially in ambiguous cases.

3.2.3 Model robustness in video detection. From the detection result provided by case A in Section 5.3, Fig. 10a and b, we can see that though the YOLO-text without prompts gives higher mAP on images, it shows an unstable detection result compared to YOLO in the video detection, which is worth studying in the future work. Additionally, one experiment could have various reaction stages that require different prompts. Fig. 8 demonstrates the detection results from YOLO-text with varying prompts provided at different experimental stages, representing a dynamic prompting scenario. This implies the importance of acquiring on-line reasoning results from the LLM, especially when the transition between different reaction stages is fast, and the fact that YOLO-text is sensitive to prompts when facing ambiguous scenarios.

4 Conclusions

We presented YOLO-text, a model that supports context-aware learning for domain-specific detection tasks with scarce data. The proposed model integrates with a pretrained LLM, which processes experiment protocols as input and outputs the expected chemical forms generated during the experiment. This LLM-generated output provides contextual information to the detection model, enhancing its predictions during inference. The contextual information aims to assist model understand ambiguous visual cues. Experimental results demonstrate that incorporating context during training improves the model's performance, particularly in detecting ambiguous areas and addressing imbalanced classes.

Future work will focus on two main areas. The first is to diversify data collection to improve data quality. We believe a larger and diverse dataset can generalize model performance on different experience setups and chemical forms. The second area is to improve the robustness of the model in real-time detection. Insights from the experimental results presented in Fig. 8–10 highlight the crucial role of providing appropriate and dynamic prompts in guiding the model's visual understanding during stage transitions. We plan to explore adaptive prompt generation mechanisms that can adjust to the experimental context in real time. One possible way to improve real-time prediction is to send video frames as feedback before the LLM and automatically detect phase changes, then force the LLM to update the reasoning result. Additionally, we plan to integrate

a knowledge reasoning module in YOLO-text to simplify the whole detection into one step, enabling YOLO-text to directly react to experiment descriptions. By achieving these goals, we aim to advance the application of computer vision technologies in monitoring chemical experiments for laboratory automation.

Author contributions

All authors conceptualized the project of introducing a context-aware computer vision model to monitor chemical experiments. The method was proposed by Junru Ren, Abhijoy Mandal, and Kourosh Darvish. Junru Ren and Abhijoy Mandal worked on method implementation, dataset organization, coding and experiments. Rama El-khawaldeh provided and annotated data. Shi Xuan Leong and Junru Ren work on prompt tuning on LLM. Junru Ren drafted the original manuscript. All authors reviewed and proofread the manuscript. Alan Aspuru-Guzik, Jason Hein, Lazaros Nalpantidis, and Kourosh Darvish supervised the project.

Conflicts of interest

There are no conflicts to declare.

Data availability

The data used to train our model originate from a subset of the open-source HeinSight 2.0, 3.0, and 4.0 datasets.[6,13,14] Each HeinSight version captures experiments of closely related types. By combining data across these different experimental setups, we created a more diverse dataset that highlights the need for context-aware modeling for generalizable models for chemical experimentation. For the specific images and annotations used in model training, please refer to the following repository: https://doi.org/10.5281/zenodo.17436705.

Supplementary information: provides details on prompt tuning and includes example outputs from the LLM used in this study. See DOI: https://doi.org/10.1039/d5dd00346f.

Acknowledgements

The authors gratefully acknowledge the research funding and resources provided by the Digital PhD Program at the Technical University of Denmark (DTU), the Acceleration Consortium at the University of Toronto, and the Matter Lab at the University of Toronto. S. X. L. acknowledges support from Nanyang Technological University, Singapore and the Ministry of Education, Singapore for the Overseas Postdoctoral Fellowship. This research is part of the University of Toronto's Acceleration Consortium, which receives funding from the CFREF-2022-00042 Canada First Research Excellence Fund. A. A.-G. thanks Anders G. Frøseth for his generous support. A. A.-G. also acknowledges the generous support of Natural Resources Canada and the Canada 150 Research Chairs program.

References

1 M. Abolhasani and E. Kumacheva, *Nat. Synth.*, 2023, **2**, 483–492.
2 M. Seifrid, R. Pollice, A. Aguilar-Granda, Z. Morgan Chan, K. Hotta, C. T. Ser, J. Vestfrid, T. C. Wu and A. Aspuru-Guzik, *Acc. Chem. Res.*, 2022, **55**, 2454–2466.
3 J. Ren, N. Fisker-Bødker, R. Güldenring, J. H. Chang, T. Vegge, O. Ravn and L. Nalpantidis, *Proceedings of 2025 IEEE/SICE International Symposium on System Integrations*, 2025.
4 N. J. Szymanski, B. Rendy, Y. Fei, R. E. Kumar, T. He, D. Milsted, M. J. McDermott, M. Gallant, E. D. Cubuk, A. Merchant, *et al.*, *Nature*, 2023, **624**, 86–91.
5 D. Schober, R. Güldenring, J. Love and L. Nalpantidis, *2025 IEEE/SICE International Symposium on System Integration (SII)*, 2025, pp. 1193–1200.
6 R. El-Khawaldeh, M. Guy, F. Bork, N. Taherimakhsousi, K. N. Jones, J. M. Hawkins, L. Han, R. P. Pritchard, B. A. Cole, S. Monfette, *et al.*, *Chem. Sci.*, 2024, **15**, 1271–1282.
7 R. El-khawaldeh, A. Mandal, N. Yoshikawa, W. Zhang, R. Corkery, P. Prieto, A. Aspuru-Guzik, K. Darvish and J. E. Hein, *Device*, 2024, **2**, 100404.
8 S. Duffield, L. Da Vià, A. C. Bellman and F. Chiti, *Org. Process Res. Dev.*, 2021, **25**, 2738–2746.
9 J. Jiang, G. Cao, J. Deng, T.-T. Do and S. Luo, *IEEE Transactions on Artificial Intelligence*, 2023.
10 C. Yan, M. Cowie, C. Howcutt, K. M. Wheelhouse, N. S. Hodnett, M. Kollie, M. Gildea, M. H. Goodfellow and M. Reid, *Chem. Sci.*, 2023, **14**, 5323–5331.
11 A. C. Sun, J. A. Jurica, H. B. Rose, G. Brito, N. R. Deprez, S. T. Grosser, A. M. Hyde, E. E. Kwan and S. Moor, *Org. Process Res. Dev.*, 2023, **27**, 1954–1964.
12 S. Eppel, H. Xu, M. Bismuth and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2020, **6**, 1743–1752.
13 R. El-khawaldeh, R. Corkery, W. Zhang, N. Khan, K. Jakuba, M. Reish, K. Jones, M. Roy, S. Monfette and J. Hein, *ChemRxiv*, 2025, DOI: 10.26434/chemrxiv-2025-sxfvl-v2.
14 K. He, G. Gkioxari, P. Dollár and R. Girshick, *Mask R-CNN*, 2018, https://arxiv.org/abs/1703.06870.
15 G. Jocher, J. Qiu and A. Chaurasia, Ultralytics YOLO, 2023, https://github.com/ultralytics/ultralytics.
16 K. Sanders, R. Kriz, A. Liu and B. Van Durme, *Adv. Neural Inf. Process. Syst.*, 2022, **35**, 2637–2650.
17 H. Chung, K. H. Park, T. Seo and S. Cho, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 444–453.
18 Y. Liu, L. Liu, Y. Guo and M. S. Lew, *Pattern Recogn.*, 2018, **84**, 51–67.
19 I. Gallo, A. Calefati and S. Nawaz, *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, 2017, pp. 36–41.
20 Y. Qin, X. Gu and Z. Tan, *Neural Netw.*, 2022, **152**, 434–449.
21 K. Tang, M. Paluri, L. Fei-Fei, R. Fergus and L. Bourdev, *Proceedings Of The IEEE International Conference On Computer Vision*, 2015, pp. 1008–1016.

22 I. Gallo, A. Calefati, S. Nawaz and M. K. Janjua, *Multimodal classification fusion in real-world scenarios published conference: 2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, 2017, DOI: 10.1109/ICDAR.2017.326.
23 A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, *International Conference On Machine Learning*, 2021, pp. 8748–8763.
24 L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10965–10975.
25 T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang and Y. Shan, et al., *arXiv*, 2023, preprint, arXiv:2303.05499, DOI: 10.48550/arXiv.2303.05499.
26 S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu *et al.*, *arXiv*, 2023, preprint arXiv:2303.05499, DOI: 10.48550/arXiv.2303.05499.
27 X. Zhao, Y. Chen, S. Xu, X. Li, X. Wang, Y. Li and H. Huang, *arXiv*, 2024, preprint, arXiv:2401.02361, DOI: 10.48550/arXiv.2401.02361.
28 G. Jocher, A. Chaurasia and J. Qiu, Ultralytics YOLO, 2023, https://github.com/ultralytics/ultralytics.
29 J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, 2016, https://arxiv.org/abs/1506.02640.
30 T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, Feature Pyramid Networks for Object Detection, 2017, https://arxiv.org/abs/1612.03144.
31 J. Pennington, R. Socher and C. Manning, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.
32 H. Zhang, *arXiv*, 2017, preprint arXiv:1710.09412, DOI: 10.48550/arXiv.1710.09412.
33 A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin and A. A. Kalinin, *Information*, 2020, **11**, 125.
34 D. Ramirez, L. J. Shaw and C. D. Collins, *Environ. Sci. Pollut. Res.*, 2021, **28**, 5867–5879.
35 T. Vander Hoogerstraete, B. Onghena and K. Binnemans, *J. Phys. Chem. Lett.*, 2013, **4**, 1659–1663.