LabAssist: Enhancing Error Detection in Chemistry Experiments using Synthetic Data for Faster and More Robust Training

SS027

Chong Choon-Hou Rafael
Ng Chen-Yi
Jerome Lim Feng

1 Background and Purpose

1.1 Introduction

Training artificial intelligence models requires the availability of large, diverse and relevant datasets for accurate results [1] [2]. However, in many real-world use cases, the collection of sufficiently varied labelled data is challenging [3], as it is expensive, time-consuming, and comes with ethical considerations that using image data of a person could entail [4]. This is especially prominent in video-based tasks, where each sample consists of hundreds of coherent frames rather than a single image [5].

In the case of video-based tasks, data augmentation [6] and synthetic data generation [7] have emerged as potential solutions to combat the challenges associated with data collection. However, conventional augmentation techniques for images like random cropping, colour jittering and transformations are limited in their ability to produce realistic and varied video data [8]. This results in augmentations falling short of improving model generalisation and convergence, especially since they fail to capture higher-level differences such as subject appearance, natural motion and background environment, all of which are essential for video model generalisation [9].

LabAssist is a computer vision system designed to detect procedural errors in chemistry titration experiments. By analysing footage of students performing titration procedures, the system aims to identify incorrect techniques and provide personalised feedback to improve learning outcomes and experimental techniques. However, collecting sufficiently diverse real-world training data is difficult due to constraints in participant availability and recording conditions [10].

This project explores the use of synthetic video data as a scalable alternative for expanding and diversifying the LabAssist training dataset [11]. We introduce variations in visual factors such as skin tone, clothing, background environment, and lighting conditions. By comparing low-fidelity and high-fidelity data generation pipelines, we aim to quantify their impacts on model performance and robustness on unseen test conditions.

Through this study, we seek to establish whether increasing the realism of synthetic data leads to consistent improvements in model generalisation and performance in the use case of LabAssist.

1.2 Problem Statement

LabAssist currently relies on manually collected video data of students conducting titration experiments. While this approach has been somewhat effective, the available footage still lacks sufficient diversity in both student appearance and environmental conditions. This limitation may reduce the model's ability to generalise to unseen real-world scenarios and result in lower model performance.

To address this issue, we propose using synthetic data to simulate a wider range of real-world conditions, including variations in skin tones, clothing styles and backgrounds. We hypothesise that introducing such variability will improve the model's robustness and accuracy in detecting procedural errors across diverse contexts.

1.3 Prior Works

The use of synthetic data in computer vision has become increasingly prevalent, particularly in domains where real annotated data is scarce or expensive to obtain [12].

Knapp et al. [13] propose a method for generating synthetic human action video data using pose transfer in order to improve model performance for action-detection models. They found that synthetic data generation improves performance in action recognition tasks and effectively scales up dataset sizes, improving the use of synthetic data to improve training effectiveness.

Li et al. [14] address the problem of data scarcity in real-world action understanding by proposing a synthetic data generation framework based on a text-to-video diffusion transformer. Their method enables the scalable generation of annotated action videos that increases the diversity for both environmental context and character appearance. Additionally, to mitigate the negative impact of low-quality generated samples, they propose an uncertainty-based label smoothing technique that reduces the influence of unreliable synthetic data during training. Their approach demonstrates strong performance across different datasets and tasks, achieving unparalleled results in zero-shot action recognition.

1.4 Hypothesis

We hypothesise that augmenting the LabAssist training dataset with synthetic titration videos containing controlled visual variations will improve the model's accuracy and robustness in detecting procedural errors [15].

2 Methodology

2.1 Overall Goal

The primary goal of this project is to investigate whether synthetic video data can meaningfully improve the generalisation and robustness of the LabAssist action classification model. Specifically, we examine whether introducing controlled visual diversity—such as changes in background, appearance, and motion characteristics—can reduce overfitting to the narrow visual distribution of real-world training data.

Model performance is evaluated under three training conditions:

- Training on real-world footage only
- Training on real-world footage combined with low-fidelity synthetic data
- Training on real-world footage combined with high-fidelity synthetic data

Beyond absolute classification accuracy, we focus on robustness to unseen test conditions, including novel student appearances and laboratory environments. We further investigate whether increasing the realism of synthetic data yields diminishing returns in downstream model performance.

2.2 Types of Detection

In this paper, we focus exclusively on the detection of the swirling action of the conical flask during the titration process. This action is a critical step in the titration process that ensures the proper mixing of the titrate and titrant.

By constraining the scope of detection to a single procedural action, we isolate the effects of synthetic data augmentation on model generalisation without accounting from differences in multiple task types. This controlled setup allows us to attribute any observed performance changes directly to the properties of the augmented data rather than to task complexity.

2.3 Data Types

We categorise our dataset into four distinct types based on the method of acquisition and degree of modification: one real-world data type, two low-fidelity synthetic data types, and one high-fidelity synthetic data type. This categorisation enables a systematic analysis of how different forms of synthetic augmentation affect model performance.

2.3.1 Real-World Footage

Real-world footage consists of videos collected from students performing titration experiments under authentic laboratory conditions. These videos are primarily front-facing and capture the natural execution of the swirling motion along with incidental variations such as hand shape, skin tone, background clutter, and lighting. This dataset represents the target distribution that LabAssist is ultimately expected to generalise to, and therefore serves as the baseline for all experiments.

2.3.2 Low-Fidelity Synthetic Footage

Low-fidelity synthetic footage is generated using lightweight, localised transformations applied to real-world videos. These transformations do not alter the underlying motion semantics of the action and are computationally inexpensive.

Background Changer. Given a real-world video sample, the pipeline performs the following steps: (1) the original background is removed using a trained segmentation model (YOLOv8-seg [16]) to isolate the hands and conical flask; and (2) the extracted foreground is composited onto new background images or videos, thereby simulating different laboratory environments.

2.3.3 High-Fidelity Synthetic Footage

High-fidelity synthetic footage is generated using generative models that perform holistic, temporally consistent modifications across frames.

Optical Flow-Based Motion Transfer. Given a set of real-world videos, the pipeline performs the following steps: (1) we fine-tune a GMFlow-based model [17] to learn dense optical flow fields that capture the motion patterns associated with the swirling action; (2) a single frame is sampled from a real-world video and passed through Google Nano Banana Pro [18] to introduce targeted visual modifications, such as changes in hand appearance, lighting conditions, or background structure; and (3) the edited frame is animated using the learned optical flow from the GMFlow model, resulting in a temporally coherent synthetic video.

Diffusion-based Image Animation. Given a real-world video sample, the high-fidelity pipeline performs the following steps: (1) a single frame is sampled from the video; (2) the sampled frame is passed through Google Nano Banana Pro [18], an image editing model, to introduce targeted visual modifications such as changes in hand appearance, lighting conditions, or background structure; and (3) the edited frame is then passed into WAN 2.2-5B, an image-to-video generation model, which animates the image into a full video while preserving the swirling motion.

This pipeline enables complex and global changes in appearance and scene structure that are not achievable through simple compositional transformations.

2.4 Low-Fidelity vs High-Fidelity Distinction

We formally distinguish between low-fidelity and high-fidelity synthetic data based on three criteria:

- **Locality**: Low-fidelity methods apply minor changes to specific parts of the video, whereas high-fidelity methods modify the entire scene.
- **Consistency**: Low-fidelity methods often apply transformations independently across frames, while high-fidelity methods force coherence across all frames.
- **Capacity**: Low-fidelity methods are compositional, reusing existing content, whereas high-fidelity methods synthesise new content through generative modelling.

2.5 Class Distribution and Balancing

The LabAssist dataset contains three class labels: *correct*, *stationary*, and *incorrect*, corresponding to proper execution of the swirling action, absence of motion, and incorrect execution respectively.

To increase data diversity, we first generate 150 synthetic videos for the *correct* class. To further mitigate class imbalance during training, we apply conventional data augmentation techniques to upsample minority classes until a uniform distribution of 300 samples per class is achieved.

2.6 Dataset Composition

The dataset is partitioned into train, validation, and test. Synthetic data is introduced only in the training and validation splits, while the test split consists exclusively of real-world footage.

Data Type	Train	Validation	Test
Real-World Footage	600	150	360
Background Changer	120	30	–
GMFlow	120	30	–
Nano Banana + WAN 2.2	120	30	–

Table 1: Dataset composition.

The test set consists exclusively of real-world footage in order to evaluate the model's ability to generalise to unseen real-world conditions. This ensures that performance metrics reflect robustness to natural variations in environment, appearance, and camera viewpoints.

2.7 Experimental Setup

We conduct controlled experiments to isolate the effect of synthetic data augmentation. Each model is trained under one of the following configurations:

1. **Real-World Data**
2. **Real-World Data with Background Changer**
3. **Real-World Data with Optical Flow**
4. **Real-World Data with Image Animation**

2.8 Training Configuration

All action-detection models share the same architecture, hyperparameters, and training schedule. The only variable is the composition of the training dataset.

We adopt a transfer learning approach using a pretrained video classification backbone. Specifically, we use the **X3D-M architecture** from the PyTorchVideo library as the base model. All convolutional layers are frozen during training, and only the final classification fully-connected network is fine-tuned.

2.9 Computational Resources and Cost Analysis

All model training and inference were performed on Nvidia A40 GPUs with a running cost of **$0.40** per hour of usage.

2.10 Nano Banana Image Alteration Cost

High-fidelity synthetic data generation was performed in two stages: generation of edited starting frames and video animation.

The starting frames were generated using the Nano Banana image editing API. Each frame required approximately **20 seconds** to generate. To improve throughput, 10 parallel API calls were used, resulting in an effective generation time of approximately 460 seconds for 233 starting frames. This gave an approximate cost per frame of **$0.039**.

2.10.1 Diffusion-based Generation Cost

After generating starting frames from Nano Banana, these images were then passed into Wan 2.2 for animation.

The Wan2.2 pipeline generated 150 videos at **15 minutes** per video, with each video being 125 frames long (approximately 5 seconds at 30 fps). This results in an estimated cost per video of $0.10 per video.

2.10.2 GMFlow Motion Transfer Cost

After generating starting frames from Nano Banana, these images were then passed into GMFlow for animation.

The GMFlow-based optical flow motion transfer pipeline generated **150 videos**, requiring **48 minutes and 50 seconds** of generation time which corresponds to an estimated cost per video of **$0.33**.

This pipeline was substantially more computationally efficient than the high-fidelity generation process while still enabling realistic motion variation through dense optical flow warping.

2.10.3 Background Replacement Cost

For the background replacement pipeline, a YOLOv8-based segmentation model was trained to isolate the hands and conical flask. The segmentation model was trained on 183 training images and 42 validation images after augmentation. Training took 18 minutes and 44 seconds, corresponding to an estimated cost per image of **$0.12**.

Manual annotation of the segmentation masks required approximately 5 hours of human labour. For consistency across augmentation pipelines, only 150 background-replaced synthetic videos were generated. The generation process took **16 minutes and 30 seconds**, corresponding to an estimated cost of **$0.11**.

2.11 Evaluation Metrics

To quantitatively assess model performance, we report standard classification metrics of ***Accuracy***, ***Precision***, ***Recall*** and ***F1-score***.

All metrics are reported using macro-averaging, such that each class contributes equally to the final score regardless of class frequency. This is particularly important in our setting, where the dataset exhibits class imbalance.

To measure robustness, we evaluate performance on a test set containing unseen visual conditions. All reported results are computed exclusively on real-world test data.

3 Results

3.1 Overall Performance Comparison

This section presents the quantitative performance of all four training configurations:

- Real-World Data
- Real-World Data with Background Changer
- Real-World Data with Optical Flow
- Real-World Data with Image Animation

Training Setup	Accuracy	Precision	Recall	F1-score
Real-World Footage	0.8333	0.4876	0.5497	0.5115
Real-World Footage + BG Change	0.2333	0.4120	0.6665	0.2243
Real-World Footage + GMFlow	0.9111	0.5657	0.5863	0.5758
Real-World Footage + Wan2.2	0.5083	0.3449	0.3709	0.3124

Table 2: Macro-averaged classification metrics across different training setups.

Table 2 summarises the macro-averaged classification performance of all four training configurations. Macro-averaging computes each metric independently for each class and then takes their unweighted mean, ensuring that all three classes—*correct*, *stationary*, and *incorrect*—contribute equally to the final score regardless of class frequency. This is particularly important in our setting, where the original dataset exhibits significant class imbalance.

3.2 Confusion Matrix Analysis

To further analyse the classification behaviour of each training configuration, we present the corresponding confusion matrices. While aggregate metrics such as accuracy and macro-averaged F1-score summarise overall performance, confusion matrices provide fine-grained insight into specific misclassification patterns between classes.

Each confusion matrix visualises the distribution of predicted labels against ground-truth labels for the three classes: *correct*, *stationary*, and *incorrect*. Diagonal entries from the top-left to bottom-right indicate correct predictions, whereas off-diagonal entries correspond to misclassifications.

True \ Predicted	Correct	Incorrect	Stationary
Correct	45	0	13
Incorrect	1	0	9
Stationary	37	0	255

Table 3: Real-world footage

True \ Predicted	Correct	Incorrect	Stationary
Correct	54	2	2
Incorrect	0	10	0
Stationary	181	91	20

Table 4: Background-changed footage

True \ Predicted	Correct	Incorrect	Stationary
Correct	46	0	12
Incorrect	5	0	5
Stationary	10	0	282

Table 5: Optical flow augmentation

True \ Predicted	Correct	Incorrect	Stationary
Correct	34	0	24
Incorrect	4	0	6
Stationary	143	0	149

Table 6: Diffusion-generated footage

4 Discussion

4.1 Effectiveness of Synthetic Data

We hypothesised that augmenting the LabAssist training dataset with synthetic titration videos containing controlled visual variations would improve the model's accuracy and robustness in detecting procedural errors. Our results partially support this hypothesis, as it indicates that the effectiveness of synthetic data on model performance is highly dependent on the augmentation pipeline.

The baseline model trained solely on real-world footage achieved an accuracy of 0.833 on the test set.

Overall, these results suggest that synthetic data can improve generalisation, but only when it preserves task-relevant semantics. Augmentations that distort core motion patterns or introduce excessive visual noise may hinder rather than help.

4.1.1 Background Changed Footage

When background-changed footage was introduced to the training set, performance dropped sharply to 0.233. This degradation suggests that not all forms of synthetic data generation are beneficial. In this case background replacement appears to introduce visual changes that disrupt the model's ability to focus on the core action dynamics of swirling.

This observation is further supported by the confusion matrix in Table 4. A substantial number of *stationary* samples were misclassified as *correct*, indicating that the model may have overfit to certain visual characteristics rather than generalising to the different actions. The increased variation in environment likely made it more challenging for the model to separate the foreground action and background elements.

4.1.2 Optical Flow Footage

In contrast, the model trained with GMFlow-based motion augmentation achieved an accuracy of 0.911, outperforming the baseline. This improvement suggests that synthetic data can enhance generalisation, especially when the underlying action can be retained. The relevant confusion matrix in Table 5 shows a reduction in

misclassification of *stationary* samples, indicating improved discrimination between motion and non-motion classes.

This is likely due to how the motion is the most preserved, allowing different environmental factors to be altered while retaining the motion to improve model generalisation. Thus, the model can be exposed to a more diverse training dataset consisting of different scene components, allowing it to learn and accurately classify the motion in the video.

This has proven to be the most reliable pipeline in extending our database size and producing high-quality synthetic data.

4.1.3 Diffusion-Generated Footage

The pipeline performed considerably worse than the baseline, achieving an accuracy of 0.5083. This is likely the result of a combination of reasons. From the videos generated, there were issues like (1) videos where the model misunderstood the prompt, animating redundant motions such as pouring liquid into the flask, and with (2) some videos bearing deformed features like conjoined fingers and contorted flasks.

This behaviour likely stems a combination of model bias, ambiguity in motion description, and limitations in how WAN 2.2 represents fine-grained procedural actions. Swirling in titration involves a very specific, controlled motion, but our prompt loosely defined the motion characteristics, such as the speed, force, rhythm and hand posture. This caused the model to hallucinate and fall back on more common action patterns.

4.2 Failure Cases and Limitations

Across all training configurations, the models exhibited persistent difficulty in distinguishing between the *correct* and *stationary* classes, particularly in cases involving subtle hand movements, minor camera shake, or background motion. These low-level temporal artefacts may resemble intentional swirling, causing false positives and suggesting that the model is sensitive to spurious motion cues rather than the semantic structure of the action. The *incorrect* class also remained challenging to classify due to its limited representation in the dataset, which restricted the model's exposure to the full diversity of failure behaviours such as irregular motion patterns or incomplete swirling cycles.

4.3 Implications for Real-World Deployment

Synthetic data generation can improve robustness to unseen real-world conditions, which is critical given the diversity of laboratory environments in deployment. This approach enables scalable data collection without extensive manual annotation.

However, ethical and practical considerations must also be addressed. Multimedia data collected by students for the generation of synthetic data must be conducted while keeping privacy and consent in mind, as well as follow moral guidelines on the use of this data to prevent misuse or the reinforcement of harmful biases.

In summary, synthetic data shows strong potential in enabling and scaling LabAssist systems, improving both the speed of data collection and robustness of model performance.

Beyond LabAssist, the findings of this work suggest that motion-preserving data generation techniques may be broadly applicable to other video-based action recognition tasks, particularly in domains where data collection is expensive, sensitive, or logistically constrained.

Appendix A Example images of data types

This section presents example frames from each of the four data types used in this study: real-world footage, background-altered footage, GMFlow-based optical flow and WAN 2.2 image-to-video synthesis.

Figure 1: *
(a) Real-world footage

Figure 2: *
(b) Background-changed footage

Figure 3: *
(c) WAN 2.2 animated footage

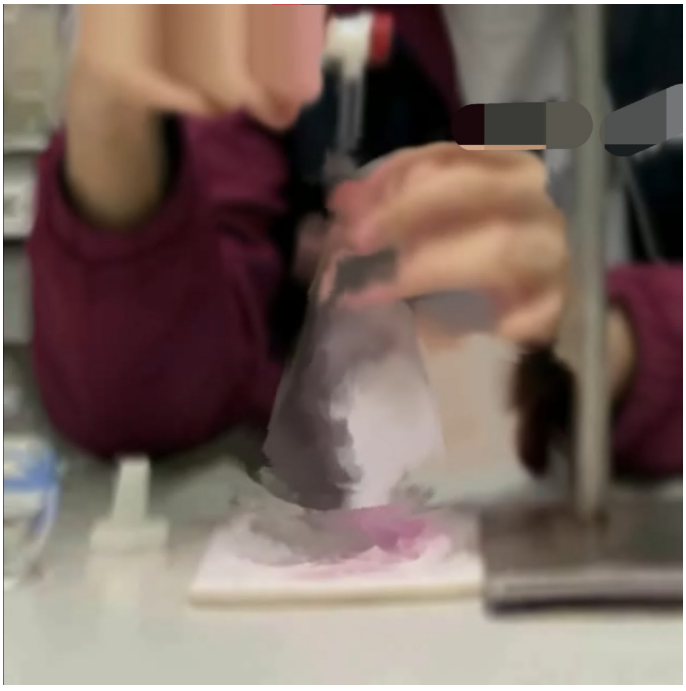

Figure 4: *
(d) Optical flow-based footage

Figure 5: Example frames from each data type used in this study

Appendix B Model Configuration

This appendix summarises the full architectural and training configuration of the action-detection model used in all experiments. These details are provided to ensure reproducibility and to clarify that all experimental comparisons were conducted under identical training conditions, with the only variable being the composition of the training data.

Parameter	**Value**
Backbone Architecture	X3D-M (PyTorchVideo)
Pretraining	Kinetics-400
Frozen Layers	All convolutional layers
Output Classes	3 (*correct*, *stationary*, *incorrect*)
Final Activation	Softmax
Loss Function	Cross-Entropy Loss
Optimizer	AdamW
Initial Learning Rate	1×10^{-3}
Learning Rate Scheduler	CosineAnnealingLR
$T_{\max}$	10
$\eta_{\min}$	1×10^{-6}
Batch Size	6
Number of Epochs	50

Table 7: Model architecture and training configuration for the action-detection network.

Appendix C Per-Class Performance Metrics

This appendix reports the per-class precision, recall, and F1-score for each training configuration, computed from the confusion matrices presented in the main paper. We additionally report the weighted averages, where each class contributes proportionally to its support (i.e., the number of ground-truth samples belonging to that class in the test set). This provides a more fine-grained view of model behaviour beyond aggregate metrics.

C.1 Real-World Footage Only

Class	Precision	Recall	F1-score	Support
Correct	0.542	0.776	0.638	58
Incorrect	0.000	0.000	0.000	10
Stationary	0.921	0.873	0.896	292
Weighted Avg.	0.834	0.833	0.830	360

Table 8: Per-class and weighted performance for the real-world-only model.

C.2 Real-World + Background Changer

Class	Precision	Recall	F1-score	Support
Correct	0.230	0.931	0.369	58
Incorrect	0.097	1.000	0.177	10
Stationary	0.909	0.068	0.127	292
Weighted Avg.	0.777	0.233	0.168	360

Table 9: Per-class and weighted performance for the background-changer model.

C.3 Real-World + GMFlow

Class	Precision	Recall	F1-score	Support
Correct	0.754	0.793	0.773	58
Incorrect	0.000	0.000	0.000	10
Stationary	0.943	0.966	0.954	292
Weighted Avg.	0.886	0.911	0.899	360

Table 10: Per-class and weighted performance for the GMFlow-based model.

C.4 Real-World + Image-to-Video (WAN 2.2)

Class	Precision	Recall	F1-score	Support
Correct	0.188	0.586	0.285	58
Incorrect	0.000	0.000	0.000	10
Stationary	0.832	0.510	0.633	292
Weighted Avg.	0.705	0.508	0.559	360

Table 11: Per-class and weighted performance for the image-to-video model.

Appendix D Failure Modes

Figure 6: (a) Deformed flask artefacts in diffusion-generated footage.

Figure 7: (b) Incorrect liquid-pouring motion hallucinated by the generative model.

References

[1] Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. On the diversity of synthetic data and its impact on training large language models, 2024.

[2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[3] Qianyu Huang and Tongfang Zhao. Data collection and labeling techniques for machine learning, 2024.

[4] Cedric Deslandes Whitney and Justin Norman. Real risks of fake data: Synthetic data, diversity-washing and consent circumvention. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1733–1744. ACM, June 2024.

[5] Balakrishnan Varadarajan, George Toderici, Sudheendra Vijayanarasimhan, and Apostol Natsev. Efficient large scale video classification, 2015.

[6] Taeoh Kim, Jinhyung Kim, Minho Shim, Sangdoo Yun, Myunggu Kang, Dongyoon Wee, and Sangyoun Lee. Exploring temporally dynamic data augmentation for video recognition, 2022.

[7] Chengxuan Li, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai. A survey on long video generation: Challenges, methods, and prospects, 2024.

[8] Nino Cauli and Diego Reforgiato Recupero. Survey on videos data augmentation for deep learning models. *Future Internet*, 14(3), 2022.

[9] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges, 2019.

[10] Fang Liu and Demosthenes Panagiotakos. Real-world data: a brief review of the methods, applications, challenges and opportunities. *BMC Med Res Methodol*, 22(1):287, November 2022.

[11] Xiao Ling, Tim Menzies, Christopher Hazard, Jack Shu, and Jacob Beel. Trading off scalability, privacy, and performance in data synthesis, 2023.

[12] Goran Paulin and Marina Ivašić-Kos. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artificial Intelligence Review*, 56, 01 2023.

[13] Vaclav Knapp and Matyas Bohacek. Synthetic human action video data generation with pose transfer, 2025.

[14] Wei Li, Dezhao Luo, Dongbao Yang, Zhenhang Li, Weiping Wang, and Yu Zhou. The role of video generation in enhancing data-limited action understanding, 2025.

[15] Ahmed Mumuni et al. A survey of synthetic data generation and augmentation in computer vision. *IEEE Access*, 2024.

[16] Ultralytics. YOLOv8 Models. `https://github.com/ultralytics/ultralytics`, 2023.

[17] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] Google. Google Nano Banana Pro image generation. `https://gemini.google/overview/image-generation/`, 2025.