FBK-NLP at ClinSkill QA 2026: Improving Temporal Reasoning via Keypoint-Augmented Inputs

Pedro Gabriel Campana[1,2], Alberto Lavelli[2], Bernardo Magnini[2],
[1]University of Padova / Via Trieste 63, Padova, Italy
[2]Fondazione Bruno Kessler / Via Sommarive 18, Povo, Trento, Italy
Correspondence: pedro.campana@phd.unipd.it

Abstract

Understanding procedural skills from visual data is a key challenge in medical AI, especially for tasks that require reasoning over temporal sequences. We report on FBK-NLP's participation at the ClinSkill QA 2026 shared task, which requires models to arrange shuffled key frames into a coherent sequence of clinical actions and provide explanations for the resulting order. We conduct a systematic study of prompting and reasoning strategies using an open and easily deployable vision-language model (VLM). The central finding of our study is that incorporating keypoint-based representations of people's body parts substantially improves temporal reasoning behind frame ordering. Furthermore, we show that model performance is highly sensitive to prompt design and to seemingly minor factors such as filename ordering and the inclusion of domain information.

1 Introduction

Recent advances in VLMs show strong capabilities in multimodal reasoning. In the medical domain, such models could meaningfully support clinical training and assessment by helping experts interpret procedural videos and verify compliance with standardized workflows. However, their behavior on structured temporal tasks remains poorly understood, as these demand not only visual recognition but also structured reasoning about how actions unfold over time. The ClinSkill QA 2026 shared task directly targets this challenge by requiring models to infer the correct chronological order of keyframes from clinical procedures.

We investigate how an open VLM can be instructed to perform temporal reasoning through systematic prompt design and input augmentation. Our approach is fully training-free, does not rely on any task-specific fine-tuning, validation, or adaptation, and reveals non-obvious factors that significantly impact performance. Our main contribution is the use of keypoint-derived predicates as a complementary source of information. They encode body dynamics (e.g., wrist positions) and provide the model with explicit cues about motion and progression. To the best of our knowledge, this is the first work to explore keypoint information in the context of keyframe ordering. We demonstrate that this approach leads to significant performance gains — a non-obvious result, as coordinate-based representations could be expected to be out-of-distribution for VLMs.

In addition, we show that seemingly minor factors — such as the alphabetical ordering of image filenames — can introduce strong biases that degrade model predictions. First, we find that increasing reasoning complexity in prompts does not necessarily improve performance and can, in fact, make models more prone to such biases. Second, we demonstrate that incorporating domain-specific few-shot examples improves robustness and ordering accuracy. Third, we explore the role of domain knowledge by rerunning our best experiments without any domain-specific information in either the prompt or the few-shot examples to explore the task's generalizability.

2 Related Work

Recent work has highlighted the difficulty that VLMs face when reasoning over ordered sequences of actions. In the textual domain, Wang et al. (2023) and Anika and Miah (2025) introduce benchmarks for step-wise and global procedural reasoning, showing how Large Language Models (LLMs) struggle to maintain correct orderings, particularly as sequence length and disorder increase. In multimodal settings, Song et al. (2025) show that models often fail to build persistent temporal representa-

Proceedings of the BioNLP 2026 (Shared Tasks), pages 92–98

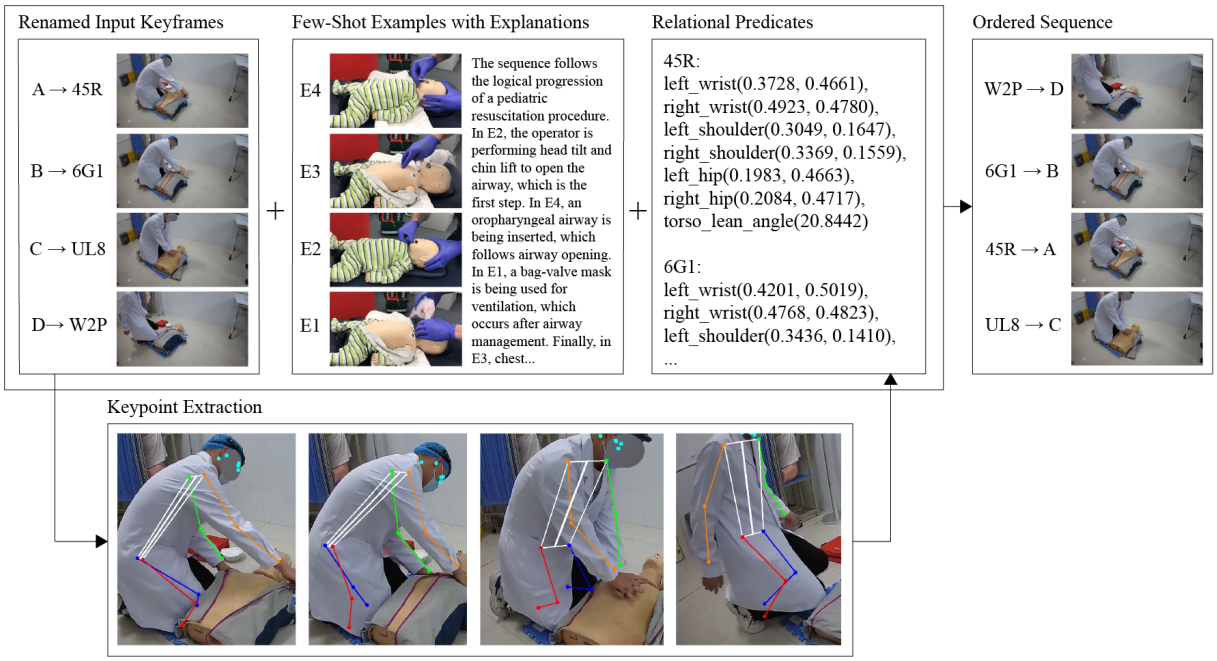

Figure 1: Overview of our most complete pipeline for clinical keyframe ordering. Unordered input frames are renamed to mitigate alphabetical bias and augmented with few-shot demonstrations. Human pose keypoints are extracted using an off-the-shelf model and converted into structured relational predicates, which are also incorporated into the prompt. The vision-language model then predicts the temporal ordering along with a textual explanation.

tions of visual sequences and instead rely on superficial cues, with performance dropping when visual information is removed.

Ma et al. (2024) also use temporal ordering as a proxy for deeper understanding in robotic settings by estimating progress from shuffled keyframes. They emphasize how ordering constraints can induce reasoning about causality, progression, and state transitions. Other studies further suggest that temporal reasoning is not only a representation issue but also a modeling and reasoning challenge: Chen et al. (2023) find that temporal information is already present in visual encodings but is not fully exploited by language models, while Ko et al. (2023) show that LLMs can exhibit temporal and causal reasoning in video QA, although in a fragile and prompt-sensitive manner.

Taken together, these works highlight both the limitations and partial successes of modern models in temporal and procedural reasoning. They motivate our focus on keyframe ordering in clinical workflows and on understanding how prompting strategies and structured representations can improve performance in a training-free setting.

3 Problem Formulation

ClinSkill QA 2026 is a shared task designed to evaluate multimodal reasoning over clinical procedures, built on the dataset introduced by Huang et al. (2026). It consists of two tasks: (1) arranging shuffled keyframes into a coherent sequence of clinical actions, and (2) providing explanations for the resulting order. Only a test set with 200 samples is provided. Each sample contains four to six keyframes depicting different stages of a basic life support (BLS) procedure. The ground-truth labels are not given to ensure fairness.

Evaluation considers both the correctness of the predicted sequence and the quality of the respective explanations. Ordering performance is measured using task accuracy, which requires exact sequence matching, and pairwise accuracy, which captures partial correctness by evaluating adjacent frame relationships. In addition, explanation quality is assessed using BERTScore and an LLM-based evaluator (G-Eval). These metrics are combined into an overall score used for leaderboard ranking. We use it as our primary metric, since the individual metrics are not provided for all submissions.

The task presents several challenges: (i) it requires relational reasoning, since the correct position of each frame depends on comparisons with the others rather than on independent classification; (ii) many temporal cues are implicit, forcing the model to infer ordering from subtle changes in posture, object configuration, or interaction dy-

namics; and (iii) the presence of textual context introduces opportunities for spurious correlations, where the model may rely on superficial signals (e.g., filename structure or prompt wording) instead of visual evidence.

4 Methodology

Our methodology examines how prompting strategies, input representations, and auxiliary cues influence zero-shot and few-shot performance in temporal ordering. Our pipeline is shown in Figure 1.

4.1 Prompting

We explore different prompting strategies, including simple formulations that request the final ordering directly and more structured prompts that require intermediate reasoning steps. They correspond to the following baselines:

Order–then–explanation. We ask the model to produce the final ordering before the explanation.

Explanation–then–order. We first ask the model for the explanations and then the ordering of the key frames, as models may revise or improve their answers during the explanation process.

Intermediate reasoning. We increase the complexity of the prompt and explicitly request intermediate reasoning steps, including frame-wise descriptions and pairwise comparisons, before the final ordering and the explanations.

4.2 Input Representation

Beyond prompting, we explore how input representations affect model predictions. In particular, we discovered that filenames can introduce unintended biases, as models may exploit alphabetical ordering as a shortcut for sequence prediction. To mitigate this effect, we use randomly assigned filenames to ablate our three initial baselines, ensuring that filenames do not encode any implicit ordering signal. This allows us to better isolate the model's reliance on visual and semantic information.

4.3 Few-Shot Examples

To provide additional task guidance, we then incorporate a few-shot demonstration into our prompts. Each demonstration consists of a set of input frames paired with correct orderings and corresponding explanations. We study both in-domain and out-of-domain demonstrations to evaluate how domain alignment affects performance, and to what extent the model can generalize from unrelated procedural contexts. More information about these examples can be found in Appendix A.

4.4 Keypoint-Derived Predicates

Our main contribution is the introduction of keypoint-derived predicates as auxiliary inputs for temporal reasoning. We extract human pose keypoints from each frame using the off-the-shelf YOLOv8 (Jocher et al., 2023) pose estimation model and convert them into structured textual descriptions. In particular, we retain the 2D coordinates of the left and right wrists, shoulders, and hips, and derive an additional torso-lean predicate from the angle between the shoulder-center and hip-center vectors (see Appendix B).

Unlike raw pixel data, these predicates offer a higher-level representation of action dynamics, encoding spatial relationships between body parts and providing explicit cues about posture and movement. The process is entirely training-free: the pose estimator is not adapted to the task, and the predicates are generated automatically without manual annotation. Furthermore, they are provided as complementary information rather than primary inputs. The model is explicitly instructed not to rely primarily on them, as keypoint extraction may be noisy or inaccurate.

5 Experiments

As a backbone VLM we use Qwen3-VL-32B-Instruct (Team, 2025) with its default hyperparameter configurations. Results can be seen in Table 1.

5.1 Prompting Baselines and Filename Bias

Among the three prompting strategies, the simple order–then–explanation approach consistently outperformed the others. In addition, intermediate reasoning yielded considerably lower scores, suggesting that more complex and demanding instructions can introduce noise and overthinking rather than improve temporal inference.

As for the filenames, random representations led to consistent improvements (approximately 3 points), confirming that models are sensitive to superficial ordering cues such as alphabetical filenames. This is supported by the fact that the number of sequences ordered as the filename alphabetic pattern, such as ABCD or ABCDEF, decreased across all baselines (see Appendix A).

Baseline	Renaming	Few-Shot Examples	Predicates	Domain Information	Score
Order–then–explanation	No	No	No	Yes	46.51
Order–then–explanation	Yes	No	No	Yes	**49.30**
Explanation–then–order	No	No	No	Yes	41.84
Explanation–then–order	Yes	No	No	Yes	45.47
Intermediate reasoning	No	No	No	Yes	39.60
Intermediate reasoning	Yes	No	No	Yes	42.93
Order–then–explanation	Yes	No	No	Yes	49.30
Order–then–explanation	Yes	Yes	No	Yes	51.13
Order–then–explanation	Yes	Yes	Yes (Subset)	Yes	56.71
Order–then–explanation	Yes	Yes	Yes	Yes	**58.79**
Order–then–explanation	Yes	No	Yes	Yes	59.45
Order–then–explanation	Yes	No	No	No	44.41
Order–then–explanation	Yes	No	Yes	No	52.34
Order–then–explanation	Yes	No	Yes	Yes	**59.75**
Order–then–explanation	Yes	Yes	No	Yes	51.13
Order–then–explanation	Yes	Yes	No	No	44.20
Order–then–explanation	Yes	Yes	Yes	No	54.47

Table 1: Summary of all experiments analyzing the impact of prompting strategy, random file renaming, few-shot demonstrations, keypoint-derived predicates, and domain information. The table is divided into three blocks: (top) comparison of prompting strategies with and without renaming, highlighting the effect of alphabetical bias; (middle) incremental improvements from few-shot examples and predicates; and (bottom) analysis of domain information and generalization.

5.2 Examples and Predicates

Adding three in-domain few-shot examples to the prompt improved the best experiment we had from 49.30 to 51.13. The gain, however, was relatively modest, suggesting that although models can benefit from a small number of task-relevant demonstrations, these provided limited additional gains for temporal ordering.

The inclusion of keypoint-derived predicates, on the other hand, led to substantial improvements. With the full predicate set, we reached a score of 58.79. We reran the experiment with only wrists and torso predicates to test whether the others introduced noise or redundancy, but this variant reached a slightly lower score of 56.71. Interestingly, our best result was obtained with the full predicate set without any few-shot examples (59.45).

5.3 Domain Information

When removing domain information only from the prompt, the score remained unchanged at 51.13, suggesting that examples can preserve domain strengths. When both the prompt and the examples were out-of-domain, however, performance dropped to 44.20, confirming that the few-shot benefit is domain-sensitive. This drop was expected, but it was interesting to see how the keypoint predicates still increased the score to 54.47, i.e., they carry meaningful task structure even without domain-specific instructions.

5.4 Discussion

Our results highlight several important aspects of temporal reasoning in VLMs beyond raw performance. First, increasing the complexity of prompting does not necessarily improve the quality of reasoning. In fact, more elaborate instructions that encourage step-by-step analysis and pairwise comparisons underperform simpler formulations. Second, random file renaming consistently improves results, revealing a non-trivial alphabetical bias in the model's behavior. Third, while in-domain examples help guide the model toward the desired output format and reasoning pattern, they are not sufficient to substantially improve performance on their own. Finally, explicit pose information provides strong guidance for temporal reasoning even when no domain information is given.

6 Conclusion

Our main contribution is the introduction of keypoint-derived predicates as structured auxiliary inputs. Extracted using an off-the-shelf pose estimator and integrated at inference time, these predicates provide explicit cues about body configuration and motion, leading to significant improvements in ordering performance.

7 Acknowledgments

This work has been partially funded by the European Union under the Horizon Europe eCREAM Project (Grant Agreement No.101057726).

References

Adrita Anika and Md Messal Monem Miah. 2025. Evaluating LLMs' reasoning over ordered procedural steps. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2259–2267.

Junwen Chen, Jie Zhu, and Yu Kong. 2023. Atm: Action temporality modeling for video question answering. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4886–4895.

Xiyang Huang, Jiawei Lin, Keying Wu, Jiaxin Huang, Kailai Yang, Renxiong Wei, Jiayi Xiang, Ziyan Kuang, Min Peng, Qianqian Xie, and 1 others. 2026. Siming-bench: Evaluating procedural correctness from continuous interactions in clinical skill videos. *arXiv preprint arXiv:2604.09037*.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. Ultralytics yolo.

Dohwan Ko, Ji Lee, Woo-Young Kang, Byungseok Roh, and Hyunwoo Kim. 2023. Large language models are temporal and causal reasoners for video question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4300–4316, Singapore. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Yecheng Jason Ma, Joey Hejna, Ayzaan Wahid, Chuyuan Fu, Dhruv Shah, Jacky Liang, Zhuo Xu, Sean Kirmani, Peng Xu, Danny Driess, Ted Xiao, Jonathan Tompson, Osbert Bastani, Dinesh Jayaraman, Wenhao Yu, Tingnan Zhang, Dorsa Sadigh, and Fei Xia. 2024. Vision language models are in-context value learners. *Preprint*, arXiv:2411.04549.

Yingjin Song, Yupei Du, Denis Paperno, and Albert Gatt. 2025. Burn after reading: Do multimodal large language models truly capture order of events in image sequences? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24316–24342.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Weizhi Wang, Hong Wang, and Xifeng Yan. 2023. Steps: A benchmark for order reasoning in sequential tasks. *arXiv preprint arXiv:2306.04441*.

A Prompting and Input Design

This appendix provides additional details to support the reproducibility of our experiments. We report the most relevant prompts used in our study, describe input preprocessing choices, and provide further information on examples and analyses.

A.1 Prompts

We report below the exact prompt used for the **order-then-explanation** baseline.

Without domain information:

```
You are analyzing a set of images depicting a sequence of actions.

You will receive several frames from the same sequence. The frames are presented in random order.

Your task is to determine the correct chronological order of the frames and then provide a short explanation describing the visual cues that determine the ordering.

Focus on cues such as changes in object positioning, visibility of elements, interaction with items, and overall progression of the scene.

In cases of ambiguity, assume that the sequence is moving forward in a natural progression, such as moving closer to a subject rather than away, or revealing more of a scene rather than concealing it.

When explaining your reasoning, include the reason for which a frame should precede or succeed the other.

You must output ONLY a valid JSON object in the following format and nothing else:

{

"predicted_order": ["<filename of the first frame in the chronological order>", "<filename of the second frame in the chronological order>", ...],

"order_rationale": "<brief explanation of the visual reasoning and progression>"

}
```

With domain information:

```
You are an expert clinician analyzing training images from a medical procedure.

You will receive several frames from the same procedure. The frames are presented in random order.

Your task is to determine the correct chronological order of the frames and then provide a short explanation describing the visual cues that determine the ordering.

Focus on cues such as operator actions, exposure of anatomical regions, hand placement, interaction with equipment, and progression of the procedure.
```

```
In cases of ambiguity, assume that the operator is progressing into the procedure, for example, moving toward the person or manikin rather than away and exposing the chest rather than covering it. Another example: a frame in which pads are already placed on the chest should occur later than frames in which they are not.

When explaining your reasoning, include the reason for which a frame should precede or succeed the other.

You must output ONLY a valid JSON object in the following format and nothing else:

{
"predicted_order": ["<filename of the first frame in the chronological order>", "<filename of the second frame in the chronological order>", ...],
"order_rationale": "<brief explanation of the visual reasoning and procedural progression>" }
```

A.2 Sequential Frame Presentation

We also experimented with an alternative prompting strategy in which frames were provided sequentially within the same conversation, rather than all at once. At each step, the model received a new frame and was expected to update its internal representation of the sequence.

This approach was motivated by the hypothesis that incremental exposure could facilitate temporal reasoning by accentuating pairwise comparisons. However, even with the keypoint-based predicates, it resulted in substantially worse performance (46.06) than all other prompting strategies. We hypothesize that this is due to the model's limited ability to maintain and revise a coherent global representation of the sequence across messages. Unlike the standard setup, where all frames are available simultaneously, this one requires persistent memory and consistent updating, which current models appear to handle poorly.

A.3 Filename Bias

We adopted the following filename-renaming scheme:

$$\begin{array}{ll} A \rightarrow 45R & D \rightarrow W2P \\ B \rightarrow 6G1 & E \rightarrow T93 \\ C \rightarrow UL8 & F \rightarrow 7WH \end{array}$$

It removes alphabetical bias while avoiding the numerical bias that would arise from renaming files to IMG_1, IMG_2, and so on. Because each sample corresponds to a different conversation, we applied the same mapping across all samples, ensuring that no filename-based patterns or associations were possible. Under this scheme, the number of sequences whose ordering matched the original alphabetical pattern (ABCD or ABCDEF) decreased as follows:

- **Order-then-explanation:** 37 → 35
- **Explanation-then-order:** 58 → 32
- **Intermediate reasoning:** 76 → 50

This indicates that, even in a purely visual task, the model may still exploit irrelevant textual structure in the input representation.

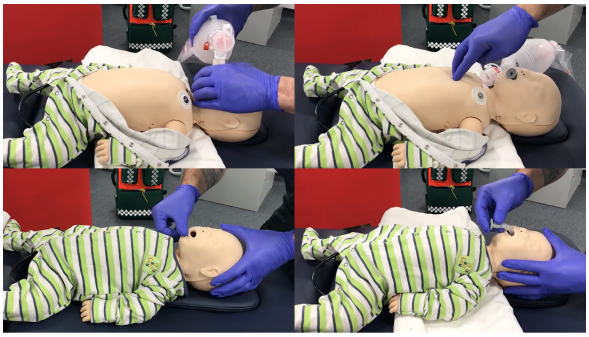

Figure 2: Example of an in-domain set of video frames collected.

Figure 3: Example of an out-of-domain set of video frames collected.

A.4 Few-Shot Examples

In-domain examples were obtained from YouTube videos by querying "basic life support" and filtering for content available under a Creative Commons license. Out-of-domain examples were obtained using the same procedure but with other queries, such as "skateboarding", "baking", and "yoga". Some of these examples can be seen in Figures 2 and 3. We always provided three examples, each presented to the model along with its correct temporal ordering and a detailed explanation written by us.

B Keypoint-Derived Predicate Construction

This appendix describes and discusses in more detail the implementation of our predicate extraction pipeline.

B.1 Pose Estimation and Keypoint Selection

The pretrained YOLOv8 pose estimator predicts the 17 COCO (Lin et al., 2014) body keypoints for every detected person. We assume the operator is the person detected with higher confidence in the frame. The detected keypoints are then normalized by image width and height to obtain scale-invariant coordinates in the range $[0, 1]$. From the complete COCO skeleton, we retain the upper-body landmarks that were empirically most informative for procedural ordering:

- left/right shoulders (indices 5 and 6),
- left/right wrists (indices 9 and 10),
- left/right hips (indices 11 and 12).

These landmarks are converted into textual predicates of the form `left_wrist(x,y)`, where x and y denote normalized image-space coordinates.

B.2 Torso Orientation

In addition to raw landmark coordinates, we derive a coarse torso orientation descriptor from shoulder and hip positions. Let

$$s = \frac{p_{ls} + p_{rs}}{2}$$

be the midpoint between the left and right shoulders, and

$$h = \frac{p_{lh} + p_{rh}}{2}$$

the midpoint between the hips. We define the torso vector as $v = s - h$. The torso lean angle is then computed relative to the upward vertical direction $u = (0, -1)$:

$$\theta = \cos^{-1}\left(\frac{v \cdot u}{\|v\|}\right).$$

The resulting value is encoded as the predicate `torso_lean_angle(`θ`)`, providing an estimate of body posture and forward leaning, which can correlate with different stages of clinical procedures.

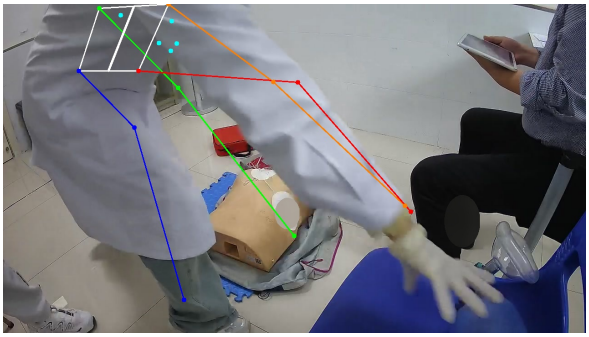

Figure 4: Example of an occluded frame where the operator is only partially captured in the camera view, resulting in an unreliable keypoint detection.

Figure 5: Example of a failure case where an observer is detected instead of the operator.

B.3 Error Analysis

As only overall submission-level scores were provided and no per-sample metrics were available, we restrict our error analysis to the keypoint-derived predicates. As expected, most errors were attributed to occlusions — both in frames where parts of the operator were deliberately masked and in others where the operator was only partially visible in the camera view. Figures 4 and 5 illustrate two representative failure cases. It would be interesting to explore whether an alternative detection strategy could mitigate them.