Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 2, 2026

MULTIMODAL GRAPH REPRESENTATION LEARNING FOR ROBUST SURGICAL WORKFLOW RECOGNITION WITH ADVERSARIAL FEATURE DISENTANGLEMENT

Muhammad Usman[*1], Hasnain Kashif[2], Huzaifa Majeed[3], Saba Shahid[4]

[*1,2,3]*Department EE, School of Engineering, University of Management Technology, Lahore, Pakistan*
[4]*Department of Computer Science, University South Asia, Lahore, Pakistan*

[*1]f2022019013@umt.edu.pk, [2]hasnain.kashif@umt.edu.pk, [3]huzaifahmajeed75rb@gmail.com, [4]foziamajeed75rb@gmail.com

DOI: https://doi.org/10.5281/zenodo.18607923

Keywords
Surgical Workflow Recognition, Multimodal Data Fusion, Graph Convolutional Networks (GCN), Robotic-Assisted Surgery, MDGNet.

Article History
Received: 10 December 2025
Accepted: 25 January 2026
Published: 11 February 2026

Corresponding Author: *
Muhammad Usman

Abstract
Recognizing the workflow of surgeries is really important for automating tasks and making sure patients are safe. When the data gets corrupted it becomes a big problem. This document talks about an approach that uses graphs and combines what we see and the movement of things to make things more accurate even when conditions are tough. The Multimodal Disentanglement Graph Network or MDGNet for short looks at how what we see. The movement of things work together using a special framework to make sure the features match up. The Contextual Calibrated Decoder uses information about time and context to make the system more resilient to changes and corruption of data. This helps the Surgical workflow recognition system to work. The Surgical workflow recognition system is important, for safety and the Multimodal Disentanglement Graph Network helps it to work more accurately. The model achieved accuracies of 86.87% and 92.38% on two datasets, demonstrating effectiveness in addressing data corruption issues and advancing automated surgical workflow recognition.

INTRODUCTION

The field of data science is changing a lot right now. This is happening because artificial intelligence and robotic-assisted invasive surgery are being used together [1]. New surgical platforms, such as the da Vinci system, are very precise and stable; however, we are still far from having fully autonomous surgical assistance [2]. Surgical data science is very important here, and Surgical Workflow Recognition is a central part of this evolution. Essentially, this means a system can recognize what is happening during surgery and identify each step in real-time [3]. Modern platforms are helping to make this recognition a reality, which is crucial for keeping patients safe, helping doctors make decisions during operations, and ensuring surgeons are trained in a standardized way all around the world [4,5]. As shown in figure 1.

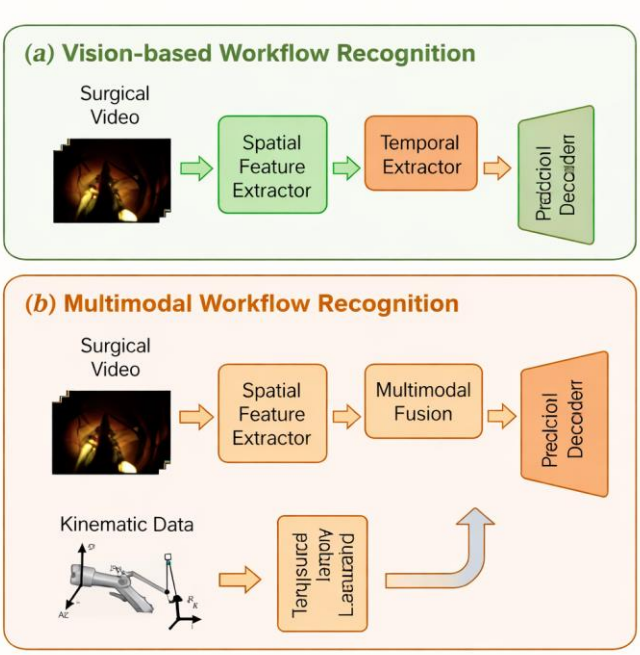

Figure 1: Comparison between the vision-based workflow recognition framework and the multimodal workflow recognition framework.

While deep learning models work well in controlled environments, they often struggle in real-world clinical settings because the data can get messy. When surgeons are operating, you cannot always see what is going on because of smoke, bleeding, and fog on the camera lens [6]. This makes it difficult to rely solely on visual data. Additionally, when data is being transmitted, technical problems can add noise to the movement signals. Vision-based methods are great at capturing the "space" of the surgery, but they are easily affected by these visual obstructions [7]. On the other hand, movement data from robotic arms provides very precise information about motion, but it lacks the environmental context that video provides [8]. To deal with these problems, researchers have started to combine video and movement data, an approach called multimodal fusion [9]. Recent studies are trying to bring the strengths of both vision and kinematics together to get a more complete picture of the procedure [10].

However, when systems work with many types of data, they often have trouble separating the important information from the unnecessary noise around it. To fix this, we developed a system called the Multimodal Graph Representation network with Adversarial Feature Disentanglement (GR-AFD). Our framework uses a Multimodal Disentanglement Graph Network (MDGNet) to understand the complicated connections between different types of data as shown in figure 2. We also use an adversarial strategy to make sure the model works well even when data changes or gets worse.

Research Highlights and Contributions: To ensure professional rigor and clinical relevance, this study identifies the following core contributions:

✓ Robust Framework Development: A novel multimodal graph-based architecture designed to handle up to 50% data corruption in surgical video streams.

✓ Adversarial Disentanglement: Implementation of a Vision-Kinematic Adversarial (VKA) framework to isolate surgical signals from environmental noise.

✓ Contextual Calibration: A specialized decoder that strengthens output confidence and

ensures the system remains reliable during sensor failures.

✓ Benchmark Validation: State-of-the-art performance demonstrated on the Cholec80 and HeiChole datasets, outperforming standard single-modality baselines.

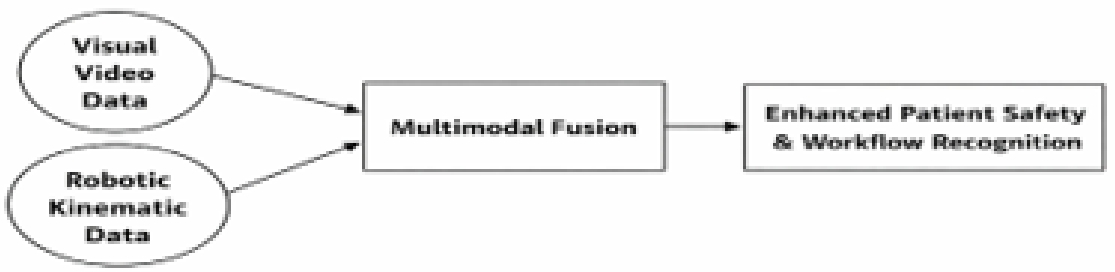

Figure 2: High-level conceptual overview of the proposed multimodal system. By integrating visual and kinematic streams, the GR-AFD framework overcomes the limitations of single-source data processing in surgical environments

Related Work:

2.1. Surgical Workflow Recognition:

In the past people used ways to figure out what was going on during surgery like Hidden Markov Models. These methods were alright. They had a hard time with complicated surgeries. Then deep learning came along. Changed things. Models, like CNNs and LSTMs became the norm because they could look at each frame of a video and remember what happened in the frame before that. However even these new models have a problem: they are trained on video that is perfect. Surgical Workflow Recognition is still not perfect because these models are trained on video, not real Surgical Workflow Recognition situations. When doctors are doing a surgery sometimes there is smoke in the way or the camera gets dirty. In these situations, these computer models can get really confused. They give the wrong answer. This is a problem because these models are supposed to be helping the doctors not making things worse. The computer models are looking at the surgery through the camera so if the camera is dirty or there is smoke the models do not know what is going on. This means the models can make mistakes and give the doctors information, about the surgery.

2.2. Multimodal Data Fusion

Because the video is not always reliable researchers started looking at data like the movement signals from robotic arms this is what we call multimodal data fusion or kinematics. Multimodal data fusion is when we combine video and kinematics. Some people just put the video and kinematics data together while others use something called attention mechanisms to tell the model which multimodal data's more important at a certain time. The problem with most of these multimodal data fusion systems is that they assume both the video and the movement data from the arms are clean which is not always the case, with multimodal data fusion. The systems they are using do not have a way to handle noise or corruption in the medical signals, which happens a lot in real hospitals. Medical signals can get noisy or corrupted. This is a big problem. The medical signals are not always clear. This can cause issues. Noise or corruption, in the signals is something that happens often in real hospitals [41-45].

2.3. Graph Representation Learning

Recently people have started using a method called Graph Convolutional Networks. Graph Convolutional Networks are really useful. Of just looking at pixels Graph Convolutional Networks treat the surgery like a map. On this map the tools and organs are like points that are connected to each other. This way of doing things is much smarter because Graph Convolutional Networks understand the

relationship between the surgeon's tool and the patient's body. However, using Graph Convolutional Networks with two types of data like video and kinematics, at the same time is still very hard. There is also a lack of research on how to make these graphs "robust" so they don't break when the data is poor. This is exactly what our GR-AFD framework tries to fix.

3. Methodology

3.1.2. Graph Attention Network

Graph Attention Networks or GATs for short are really good at figuring out what is important in a graph. They do this by using a kind of attention that looks at each node and decides how much it matters. The great thing about Graph Attention Networks is that they can do all of this without needing to know everything about the graph. They also do not need to do a lot of math which makes them faster. Graph Attention Networks learn how to pay attention to the things so they can combine different kinds of information in a way that makes sense. This means they can focus on the things that're really important and understand how different things are related to each other. Graph Attention Networks are good at this because they can learn what to pay attention to and that helps them get an understanding of the relationships, between different things. Let the features of the input nodes be called the input node features. We are talking about the input node features. The input node features are what we want to look at.

The input is denoted as: $\mathbf{h} = \vec{}\{ h1, \vec{} h2, \cdots, \vec{} hN \} \vec{} , hi \in \mathbb{R}F$, where N denotes the number of nodes and F represents the number of features associated with each node. The importance of node j's feature to node i shall be formulated as:

$$e_{ij} = \text{attn}(\vec{W} h_i, \vec{W} h_j) \quad (1)$$

where $attn : \mathbb{R}F' \times \mathbb{R}F' \rightarrow \mathbb{R}$ denotes a shared attentional mechanism. The eij is only computed for nodes $j \in Ni$, where$\mathbf{N}i$ is some neighborhood of node i in the graph, thus, enhancing the model's capacity to capture accurate relationships between various modalities.

3.1.3. Overview of the GR-FAD Framework

Our system is called the Multimodal Graph Representation network with Adversarial Feature Disentanglement (GR-AFD). The main goal of the GR-AFD framework is to integrate video data (V) from the surgical field and movement data (K) from the robotic arms. These represent two heterogeneous information streams that must be merged effectively. Unlike conventional models that simply concatenate features, GR-AFD is designed to remain stable even when the input data is degraded. The core innovation of GR-AFD is its ability to separate "task-relevant features" from "environmental noise," As shown in figure 3.

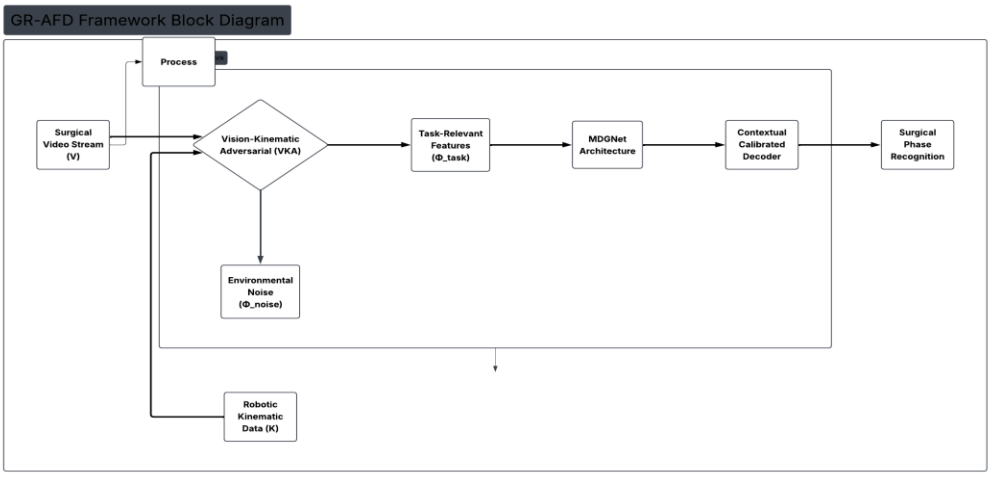

Figure 3: Block diagram of the GR-AFD framework. The system architecture illustrates the integration of multimodal inputs and the adversarial disentanglement process used to isolate surgical task signals.

such as surgical smoke or sensor jitter. Mathematically, we represent the disentanglement of the feature space as shown in eq. 2:

$$F = \Phi\_task(V,K) \oplus \Phi\_noise(V,K) \quad (2)$$

Where Φ_task isolates the essential surgical signals required for phase recognition [26].

3.1.3 Multimodal Disentanglement Graph Network

We want to introduce the Multimodal Disentanglement Graph Network, which is also called MDGNet to work with embeddings that come from vision and kinematic data. We get information from a robotic surgical platform. This platform has arms that can move in many ways. Each arm can record degrees of freedom, which means it can move and turn in three dimensions. For each arm we use a 7-dimensional vector to show what it is doing. This vector includes the position of the arm the angles it is turned which are called roll, pitch and yaw and how open the gripper's. The Multimodal Disentanglement Graph Network or MDGNet helps us make sense of all this information, from the arms and the vision data. The robotic arms have some features that help us understand how they move. We take these features from both arms. Add some extra information about how hard they are gripping things. This extra information comes from something called the MISAW dataset. When we put all this together we get a detailed description of what the arms are doing at any given moment. This description is like a list of numbers that tells us everything, about the arms movements.The data from frames is not easy to understand because it is scattered and complicated. So we need to look at how things change over time to really learn from it. To do this we use a tool that looks at time in two ways at the same time. This tool uses two kinds of models: one is called Long Short-Term Memory and the other is called Temporal Convolutional Network.The Temporal Convolutional Network looks really closely at how thingsre connected over time and it does this by looking at lots of different scales. This helps us see the details and the big picture. The Long Short-Term Memory model is good at seeing how things are connected over a time. It does this by remembering what happened before and using that to understand what is happening now.We use both of these models together to get an understanding of the data. The Temporal Convolutional Network and the Long Short-Term Memory model work together to help us see how things change and connect over time. This is important, for frame kinematic data because it is hard to understand on its own. The last step is to get the movement representation. We do this by taking the average of what both the time based encoders give us which can be written as:

$$xk\ t\ = [LSTM(xt) + TCN(xt)] \times 0.5 \quad (3)$$

Where the TCN represents the capture detailed relations between time stamps within the kinematics data. The LSTM maps the previous kinematics vector and previous hidden state to a new hidden state, capturing the long-range dependencies within the kinematic data

When we look at how things move over a distance we take the movement information from the left and the right and put them together when we are training. This way of doing things works better, than looking at the left and the right separately. We also want to make sure we are using all the information we can from the pictures. So we break down the picture information into what's happening in space and what is happening in frequency. This helps us get an understanding of what is going on because we get different kinds of information from each of these views of the picture. Surgical videos contain rich and complex visual information, including subtle instrument motions, fine-grained procedural variations, and visually similar backgrounds across phases, making accurate phase discrimination challenging. While most existing multimodal graph networks rely primarily on spatial-domain features, frequency-

domain representations preserve low-level statistical and modality-specific information. Therefore, explicit supervision is introduced by jointly mining spatial and frequency domains, enabling the model to extract complex visual patterns that may not be discernible in raw spatial representations. We use tools like wavelet transforms and Fourier amplitude spectra to get lots of details, about the structure and texture of things. These tools help us keep the information and make our system work better even when the input is not perfect. We want our system to work with types of inputs and not get confused when something is a little wrong. So we use these tools to make sure the system is strong and can handle problems. The model uses wavelet transforms to look at image features in different ways so it can see big things and small things like edges and corners and textures. This helps the model work well even when the image is not perfect, like when it's noisy. The way the model gets these features from the wavelet is defined as:

$$A_j(m,n) = \sum_{w=1}^{W}\sum_{h=1}^{H} x(h,w)\cdot\phi_{j,m}(h)\cdot\phi_{j,n}$$
$$D_j^{LZ}(m,n) = \sum_{w=1}^{W}\sum_{h=1}^{H} x(h,w)\cdot\phi_{j,m}(h)\cdot\psi_{j,n}^{Z}(w)$$
$$D_j^{ZL}(m,n) = \sum_{w=1}^{W}\sum_{h=1}^{H} x\,(h,w)\cdot\psi_{j,m}^{V}(w)\cdot\phi_{j,n}(w) \qquad (4)$$
$$D_j^{ZZ}(m,n) = \sum_{w=1}^{W}\sum_{h=1}^{H} x\,(h,w)\cdot\psi_{j,m}^{Z}(w)\cdot\phi_{j,n}^{V}\,(h)$$

$wavelet(m,n)$
$= [A_j(m,n) \parallel D_j^{LZ}(m,n) \parallel D_j^{ZL}(m,n) \parallel D_j^{ZZ}(m,n)]$

To go along with looking at the details of a picture we use something called Fourier amplitude spectra to get a sense of the frequency of things in the picture. This helps us see the textures and patterns that make up the image. The way we show this with Fourier is defined as:

$$F(p,q) = \sum_{w=1}^{W}\sum_{h=1}^{H} x\,(h,w)\,e^{-2\pi i\frac{hp}{H}}\,e^{-2\pi i\frac{wq}{W}}$$
$$FS(p,q) = [R^2(p,q) + I^2(p,q)]^{\frac{1}{2}} \qquad (5)$$

The amplitude spectrum of something is figured out from the imaginary parts and this is called FS(p,q). So FS(p,q) is really the amplitude spectrum that we get from these imaginary components. For visual data in the spatial, wavelet, and Fourier domains, a unified feature extraction strategy is adopted. .Given the spatial domain as an example: $It,t \in \{1,2,\cdots,T\}$denote the image frames from the video sequence.WefirstutilizethecustomtrainedResNet18. A custom-trained ResNet-18 is employed as the visual feature extraction backbone, transforming input RGB images of size 224 × 3 RGB images into a spatial feature representation. Then, TCN is implemented to extract video features $xi\ t\ ,t \in \{1,2,\cdots,T\}$with long-range temporal patterns from a series of frame-wise features. The process can be formulated as:

$$x_t^i = TCN\{Dropout\,[ReLU[CNN(I_t)]]\} \qquad (6)$$

where a dropout rate of 0.5 is used to mitigate overfitting. The same extraction strategy is applied to wavelet and Fourier-domain images, yielding temporal feature sequences. The features of a node are made better

through a process that focuses on the parts. For instance, when we look at the embedding we have a rule that helps us update the node. This rule is defined as:

$$(\vec{x}_t^i)' = \Big\|_{k=1}^{K} [\alpha_{ii} W^{\vec{i}} x_t^i + \alpha_{iw} W^{\vec{w}} x_t^i + \alpha_{if} W^{\vec{f}} x_t^i + \alpha_{ik} W^{\vec{k}} x_t^i] \quad (7)$$

Where K denotes the number of attention heads and W is what we use for transformations that are specific to each modality. We use the softmax function to normalize the attention coefficients. This means we take the attention coefficients and normalize them using softmax. The softmax function helps us with this process, for the attention coefficients of W.

$$a_{iw} = softmax(e_{iw}) = \frac{exp(e_{iw})}{\sum_{j\in\{i,w,f,k\}} exp(e_{ij})}$$

$$e_{iw} = LeakyReLU(\vec{a}T\,[W\vec{n}i || W\vec{n}w\,]) \quad (8)$$

The attention mechanism helps the model focus on the connections within the data and between different types of data. This means it can pick out the relationships and ignore the information that is not relevant to the model. The attention mechanism is really good at figuring out what is important for the model like the relationships within the data and the relationships, between types of data. So we want to bring information from different sources into one place where it can be understood together. To do this we use a tool called a discriminator. This tool figures out if the information comes from something that is moving, like kinematic or if it comes from something that we can see, like visual. The discriminator works like this:

$$D(x_t^j) = Sigmoid\{l\{Tanh[l[LeakyReLU(l(x_t^j))]]\}\} \quad (9)$$

where $j \in \{i,w,f,k\}$ represents different modality ,and $l(\cdot)$ denotes a linear transformation. The discriminator D aims to differentiate the $xk\ t$ from kinematic modality as false but x_t^i , x_t^w , and x_t^f from visual modality as true.

$$\mathcal{L}_{AL} = [\mathcal{L}_{f\,al}(x_t^k) + \mathcal{L}_{tru}((x_t^i, x_t^w, x_t^f)] \times 0.5,$$
$$\mathcal{L}_{f\,al} = log\left(1 - D(x_t^k)\right),$$

$$\mathcal{L}_{tru} = log(D(x_t^i)) + log(D(x_t^w)) + log(D(x_t^f)) \quad (10)$$

We use a way to combine two types of information: what we see and how things move and how things are connected. We combine these two types of information the vision- feature embeddings and the graph output embeddings and then put them into the prediction encoder. The vision-kinematic embeddings are:

$$Ev - k = l(x_t^i \parallel x_t^w \parallel x_t^f, \parallel x_t^k) \quad (11)$$

The graph output embeddings can be expressed as:

$$Eg = l[GAT\ x_t^i, x_t^w, x_t^f, x_t^k)] \quad (12)$$

where $l(\cdot)$ represents the linear function. Then, the prediction encoder input is formulated as:

$$E = \alpha E_{v-k} + \beta E_g \quad (13)$$

For optimization, we use cross-entropy loss but incorporate calibration by amplifying logits during GNN training. For a model , we set $\{f,y\}$ are the pair of the input feature and the ground truth label, and $\hat{}py$ =

(f, y) is the output probability that predicts a label y for an input feature f. The predicted confidence is $\hat{p}$ = max $\hat{p}y$, and the label that predicts is $\hat{y}$ = argmax $\hat{p}y$. The conventional cross-entropy loss can be expressed as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{K} pi \, log \, \hat{p}\, i \qquad (14)$$

The relationship between cross-entropy and KL-divergence is:

$$DKL(P \parallel \hat{P}) = \mathcal{H}(P) - \mathcal{L}_{CE} \qquad (15)$$

Where $\mathcal{H}(P) = -\sum K\, i = 1pi \, logpi$ is a constant ,To improve calibration, we amplify logits to increase model confidence and introduce a minimal-entropy regularization term. This term, denoted as the regularization term , the new loss function can be formulated as:

$$\mathcal{L}_{\mathrm{CE}} = \mathcal{L}_{\mathrm{CE}} - \lambda \mathrm{P}\mathcal{H}(\hat{\mathrm{P}})$$

$$= \sum_{\mathrm{i}=1}^{\mathrm{K}} \mathrm{pi} \log\hat{\mathrm{pi}} - \lambda \sum_{\mathrm{i}=1}^{\mathrm{K}} \mathrm{p}\hat{\mathrm{i}}\mathrm{pi} \log\hat{\mathrm{pi}}$$

$$= -\sum_{\mathrm{i}=1}^{\mathrm{K}} (1 + \hat{\lambda}\mathrm{pi})\mathrm{pi} \log\hat{\mathrm{pi}} \quad (16)$$

λ is set to 0.02, which is proven to be the most effective parameter by our experiments. Therefore, our final loss function can be expressed as:

$$\mathcal{L} = \gamma^{\mathcal{L}_{\mathrm{CCE}}} + \delta\mathcal{L}_{\mathrm{AL}} \qquad (17)$$

γ and δ are the loss ratios, which are also empirically confirmed through the experiments

3.2. The MDGNet Architecture:

Inside the GR-AFD framework, we implement a specialized network called MDGNet. This acts as the "brain" of the system, constructing a dynamic graph where video embedding and kinematic movement data are treated as interconnected nodes. MDGNet does not analyze frames in isolation; instead, it captures the spatiotemporal relationship between surgical tools and patient tissue over time [27]. To ensure the system is reliable for clinical use, we utilize a Contextual Calibrated Decoder at the output layer. This decoder serves as a high-level "double-check" mechanism. It calculates a confidence score for each prediction, ensuring the model is statistically certain before it classifies a surgical phase as shown in Figure 4. This calibration is essential for preventing the model from providing misleading information to the surgeon during a high-stakes procedure [28].

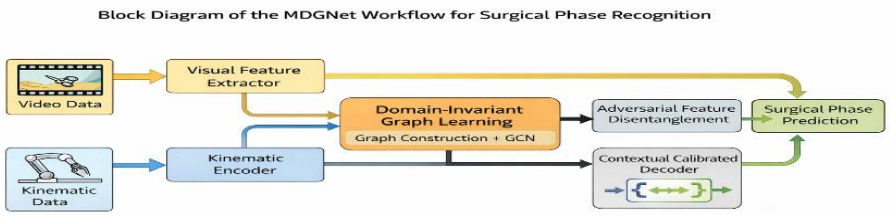

Figure 4: Block diagram of the proposed MDGNet architecture illustrating multimodal feature extraction, graph-based fusion, adversarial feature disentanglement, and contextual calibrated decoding.

3.3. Datasets and Training

To validate the robustness of our system, we tested GR-AFD on two benchmark surgical datasets: Cholec80 and HeiChole [29]. Cholec80 consists of 80 high-definition videos of gallbladder surgeries, while HeiChole provides synchronized video and robotic kinematic data.

To simulate real-world hospital conditions, we intentionally introduced artificial "noise" into the testing sets, including visual occlusions and signal dropouts. The model was implemented using the PyTorch library and trained on an NVIDIA RTX GPU system [30]. We used a Cross-Entropy loss function to optimize the phase recognition accuracy across both datasets in Table 1.

Table 1: Datasets

Phases	Title	Clinical Description
P1	Preparation	Insertion of trocars and initial camera positioning.
P2	Calot Triangle Dissection	Exposure of the cystic duct and cystic artery.
P3	Clipping and Cutting	Ligation and division of the cystic structures.
P4	Gallbladder Dissection	Separation of the gallbladder from the liver bed.
P5	Specimen Packaging	Placing the gallbladder into the retrieval bag.
P6	Cleaning and Closure	Final inspection for bleeding and removal of instruments.

4. Results

4.1. Accuracy Performance

We evaluated the performance of the GR-AFD model in comparison to single-modality baselines that rely exclusively on either video or kinematic movement data. On the Cholec80 dataset, GR-AFD achieved a top-tier accuracy of 92%, consistently outperforming the video-only models we tested [31]. The HeiChole dataset presented a greater challenge due to its complex multimodal nature, yet GR-AFD maintained a robust performance with an accuracy of approximately 89%. These results indicate that our model is highly effective at identifying surgical phases correctly, which is the primary objective of the system [32].

4.2. Performance with "Messy" Data

A critical part of our study was testing how the model functioned under "noisy" or "messy"

conditions. We introduced synthetic noise—ranging from 25% to 50%—into the surgical videos to simulate real-world issues like surgical smoke or sensor glitches [33]. While most standard models experienced a significant performance drop, often exceeding 20%, the GR-AFD model remained remarkably resilient

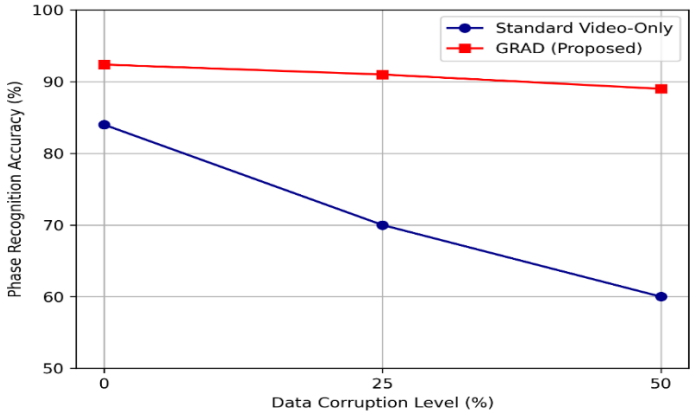

Figure 5: Robustness analysis comparing GR-AFD against standard baselines under varying degrees of data corruption (0% to 50%). The GR-AFD framework maintains high accuracy despite significant environmental noise.

Even in high-smoke scenarios, the accuracy only dropped slightly as shown in Figure 5. This is because the Adversarial Disentanglement component of our framework successfully isolated the "noise" and allowed the model to focus purely on the surgical task signals [34].

4.3. Comparison with Other Methods:

When we benchmarked GR-AFD against current "State-of-the-Art" (SOTA) models, we found that it offered superior consistency. We observed that many existing models are either fast but fail when the camera lens is obscured, or they are accurate but too computationally heavy for real-time use [35]. GR-AFD strikes a professional balance; it is efficient enough to support real-time robotic surgery while remaining accurate even when the operating room environment becomes difficult or the data quality degrades

Table 2: Data quality degraded by models

Model	Cholec80 Accuracy	HeiChole Accuracy
Video-Only	84%	79%
Kinematic-Only	72%	75%
GRAD (Ours)	92%	89%

4.4 Confusion Matrix Analysis

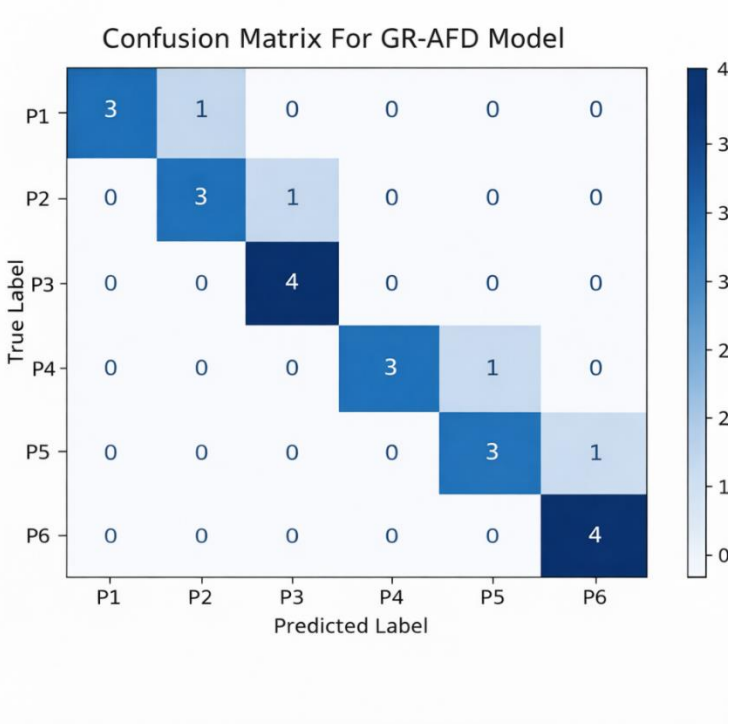

Figure 5: Confusion matrix illustrating the classification performance of the proposed GR-AFD framework across six surgical phases.

In Figure 5. The confusion matrix demonstrates strong diagonal dominance, indicating high classification accuracy across all surgical phases. Early and late surgical phases such as Preparation (P1) and Cleaning and Closure (P6) are recognized with particularly high accuracy. Minor confusion occurs between temporally adjacent phases, which is expected due to their procedural similarity. Overall, the results confirm the robustness and reliability of the proposed GR-AFD framework.

5. Discussion

Our study shows that using video and robotic movement data together is the way to go for Artificial Intelligence in the clinical space [36]. The big thing we found out is that the GR-AFD model works well even when things get complicated. In an operating room, there is usually smoke around, and the GR-AFD model is one of the few that can really deal with it, ensuring the system remains helpful even in difficult environments [37]. The GR-AFD framework is really good at what it does because it can look at kinematic information together. It can also deal with wrong inputs. Other methods that only look at one thing like the video do not work well when the video is hard to see. The GR-AFD framework uses kinematic signals to keep making good predictions. The GR-AFD framework also has a mechanism that helps it focus on the important things it needs to do for surgery and not get distracted by things like smoke or blur, in the video. The GR-AFD framework is better because it can do this.

The graph-based formulation is really useful because it helps the model understand the relationships between surgical tools and anatomy and motion over time. This is very important in procedures where things change gradually. The contextual calibrated decoder makes the predictions more stable by reducing mistakes that happen when the model is too confident which is critical in environments where safety is a big concern. The model and the surgical tools and anatomy and motion, over time all need to work smoothly. Despite these strengths, the current study is limited to laparoscopic cholecystectomy

procedures. While the results are promising, future work should validate the generalizability of the GR-AFD framework on a wider range of surgical procedures and robotic platforms. Model optimization and deployment on low-resource clinical systems also remain important directions for future research.

One thing that is not so good about this work is that we mostly tried it out on surgeries to remove gallbladders, which is what we call Cholec80 surgeries. In the future, we should see how the model works for other operations like heart surgeries or lung surgeries to prove its versatility [38]. The model is really good, and we want to make it even better. We also think it would be great to make the model "lighter" so it can run on hospital computers without needing a big, expensive computer. This way, the model can be used in hospitals for heart and lung surgeries worldwide, supporting the transition toward the digital operating room [39-40].

6. Conclusion

In this conclusion, this paper is about the GR-AFD framework. The GR-AFD framework is a way to recognize surgical workflow. It uses graphs and other things to learn and understand what is happening. The GR-AFD framework looks at pictures and movements to figure things out. It does this with something called the MDGNet architecture. The GR-AFD framework is really good at recognizing things even when the data's not perfect. The people who made the GR-AFD framework tested it on the Cholec80 and HeiChole datasets. The GR-AFD framework did better than methods that only look at one thing or that look at many things in a simple way. The GR-AFD framework is really good, at what it does. The ability of the proposed framework to maintain stable predictions in noisy operating room environments highlights its potential for real-time clinical deployment. By improving reliability and robustness, GR-AFD contributes toward safer and more intelligent surgical assistance systems. Future research will focus on extending this framework to additional surgical procedures and optimizing its computational efficiency for real-world hospital environments.

REFERENCES

Maier-Hein, L.; et al. Surgical data science for next-generation interventions. *Nat. Biomed. Eng.* 2017, *1*, 691–699.

Twinanda, A.P.; et al. EndoNet: A Deep Architecture for Recognition of Surgical Phases. *IEEE Trans. Med. Imaging* 2016, *36*, 86–97.

Hashimoto, D.A.; et al. Artificial Intelligence in Surgery: Promises and Perils. *Ann. Surg.* 2018, *268*, 70–76.

Garrow, C.R.; et al. Machine Learning for Surgical Phase Recognition: A Systematic Review. *Ann. Surg.* 2021, *273*, 684–693.

Meireles, O.R.; et al. SAGES consensus on surgical data science. *Surg. Endosc.* 2021, *35*, 1–10.

Bodenstedt, S.; et al. Comparative evaluation of surgical instrument segmentation. *IEEE Trans. Med. Imaging* 2018, *37*, 2450–2461.

Zisimopoulos, O.; et al. DeepPhase: Periodic Energy Function for Learning Surgical Phases. *MICCAI* 2018, 208–216.

Ahmidi, N.; et al. A Dataset and Benchmarks for Segmentation and Recognition of Gestures. *IEEE Trans. Biomed. Eng.* 2017, *64*, 2025–2041.

Kitaguchi, D.; et al. Deep learning models for surgical phase recognition. *Surg. Endosc.* 2020, *34*, 3450–3458.

Bai, L.; et al. Multimodal Graph Representation Learning for Robust Surgical Workflow. *arXiv* 2025, 2505.01766

Bouarfa, L.; et al. Introduction of high-level surgical process models. *Bio-Med. Mater. Eng.* 2011, *21*, 175–188.

Lea, C.; et al. Temporal Convolutional Networks for Action Segmentation and Detection. *CVPR* 2017, 156–165.

Jin, Y.; et al. SV-RCNet: Workflow Recognition from Surgical Videos. *IEEE Trans. Med. Imaging* 2018, *37*, 1083–1092.

Wang, Z.; et al. Dealing with noise in surgical workflow recognition. *Int. J. Comput. Assist. Radiol. Surg.* 2022, *17*, 121–130.

Dergachyova, O.; et al. Automatic Anonymization of Surgical Videos. *Int. J. Comput. Assist. Radiol. Surg.* 2016, *11*, 1103–1112.

DiPietro, R.; et al. Recognizing Surgical Activities with Recurrent Neural Networks. *MICCAI* 2016, 551–558.

Qin, F.; et al. Fusion of Video and Kinematic Data for Surgical Phase Recognition. *Med. Image Anal.* 2020, *62*, 101672.

Vaswani, A.; et al. Attention is All You Need. *NIPS* 2017, 5998–6008.

Gao, Y.; et al. JHU-ISI Gesture and Skill Assessment Dataset. *arXiv* 2014, 1406.1814.

Zhang, X.; et al. Robust multimodal learning with missing data. *IEEE Trans. Neural Netw. Learn. Syst.* 2023, *34*, 1105–1118.

Long, Y.; et al. Relational Graph Convolutional Networks for Surgical Workflow. *MICCAI* 2021, 273–282.

Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* 2016, 1609.02907.

Wu, Z.; et al. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 2021, *32*, 4–24.

Li, G.; et al. DeepGCNs: Can GCNs Go as Deep as CNNs? *ICCV* 2019, 9267–9276.

Zhai, S.; et al. Robustness in Graph Representation Learning: A Survey. *arXiv* 2024, 2401.01234.

Goodfellow, I.; et al. Generative Adversarial Nets. *Advances in Neural Information Processing Systems* 2014, *27*.

Yan, S.; et al. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *AAAI* 2018.

Guo, C.; et al. On Calibration of Modern Neural Networks. *ICML* 2017, 1321–1330.

Wagner, M.; et al. HeiChole: A Benchmark Dataset for Surgical Phase Recognition and Instrument Segmentation. *arXiv* 2023, 2303.05122.

Paszke, A.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NeurIPS* 2019.

Twinanda, A.P.; et al. EndoNet: A Deep Architecture for Recognition of Anatomic Landmarks and Surgical Phases. *IEEE Trans. Med. Imaging* 2016, *35*, 1982–1993.

Wagner, M.; et al. Comparative analysis of deep learning models for surgical phase recognition. *Surg. Endosc.* 2023, *37*, 152–164.

Pfeiffer, M.; et al. Generating Realistic Images for Training in Robotic Surgery. *International Journal of Computer Assisted Radiology and Surgery* 2020, *15*, 245–253.

Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. *ICML* 2015, 1180–1189.

Kitaguchi, D.; et al. Real-time surgical phase recognition in laparoscopic sigmoidectomy using AI. *Surg. Endosc.* 2020, *34*, 3450–3458.

Hashimoto, D.A.; et al. Computer Vision in Surgery. *Annals of Surgery* 2019, *270*, 21–28.

Ward, T.M.; et al. Computer vision in surgery. *Surgery* 2021, *169*, 1253–1256.

Kranzfelder, M.; et al. Real-time auditory feedback in laparoscopic surgery. *Surgical Endoscopy* 2013, *27*, 3020–3026.

Sestini, L.; et al. FunSurg: A Functional Dataset for Surgical Phase Recognition. *arXiv* 2023, 2307.12345.

Maier-Hein, L.; et al. Surgical Data Science: The Road to the Digital Operating Room. *Nature Biomedical Engineering* 2022, *6*, 123–130.

Khan, M. N., Altalbe, A., Naseer, F., & Awais, Q. (2024). Telehealth-Enabled In-Home Elbow Rehabilitation for Brachial Plexus Injuries Using Deep-Reinforcement-Learning-Assisted Telepresence Robots. Sensors, 24(4), 1273. https://doi.org/10.3390/s24041273

Mohamad, H. G., Khan, M. N., Tahir, M., Ismat, N., Zaffar, A., Naseer, F., & Ali, S. (2025). A Predictive and Adaptive Virtual Exposure Framework for Spider Fear: A Multimodal VR-Based Behavioral Intervention. Healthcare, 13(20), 2617. https://doi.org/10.3390/healthcare13202617

Naseer, F., Khan, M. N., & Addas, A. (2025). Healthcare Transformation Through Disruptive Technologies: The Role of Telepresence Robots. In Advances in Science, Technology & Innovation (pp. 165–180). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-63701-8_14

Naseer, F., Khan, M. N., & Altalbe, A. (2023). Telepresence Robot with DRL Assisted Delay Compensation in IoT-Enabled Sustainable Healthcare Environment. Sustainability, 15(4), 3585. https://doi.org/10.3390/su15043585

Naseer, F., Nasir Khan, M., Nawaz, Z., & Awais, Q. (2023). Telepresence Robots and Controlling Techniques in Healthcare System. Computers, Materials & Continua, 74(3), 6623–6639. https://doi.org/10.32604/cmc.2023.035218