On the pitfalls of Batch Normalization for end-to-end video learning: A study on surgical workflow analysis

Dominik Rivoir *, Isabel Funke, Stefanie Speidel

Department of Translational Surgical Oncology, National Center for Tumor Diseases (NCT/UCC Dresden), Fetscherstraße 74, 01307 Dresden, Germany; German Cancer Research Center (DKFZ), Heidelberg, Germany; Faculty of Medicine and University Hospital Carl Gustav Carus, TUD Dresden University of Technology, Dresden, Germany; Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Dresden, Germany
Centre for Tactile Internet with Human-in-the-Loop (CeTI), TUD Dresden University of Technology, Dresden, Germany

ARTICLE INFO

ABSTRACT

Batch Normalization's (BN) unique property of depending on other samples in a batch is known to cause problems in several tasks, including sequence modeling. Yet, BN-related issues are hardly studied for long video understanding, despite the ubiquitous use of BN in CNNs (Convolutional Neural Networks) for feature extraction. Especially in surgical workflow analysis, where the lack of pretrained feature extractors has led to complex, multi-stage training pipelines, limited awareness of BN issues may have hidden the benefits of training CNNs and temporal models end to end. In this paper, we analyze pitfalls of BN in video learning, including issues specific to online tasks such as a 'cheating' effect in anticipation. We observe that BN's properties create major obstacles for end-to-end learning. However, using BN-free backbones, even simple CNN–LSTMs beat the state of the art on three surgical workflow benchmarks by utilizing adequate end-to-end training strategies which maximize temporal context. We conclude that awareness of BN's pitfalls is crucial for effective end-to-end learning in surgical tasks. By reproducing results on natural-video datasets, we hope our insights will benefit other areas of video learning as well. Code is available at: https://gitlab.com/nct_tso_public/pitfalls_bn.

1. Introduction

Batch Normalization (BatchNorm/BN) (Ioffe and Szegedy, 2015) is a highly effective regularizer in visual recognition tasks and is ubiquitous in modern Convolutional Neural Networks (CNNs) (He et al., 2016; Szegedy et al., 2016; Tan and Le, 2019). It is, however, also considered a source for silent performance drops and bugs due to its unique property of depending on other samples in the batch and the assumptions tied to this (Brock et al., 2021; Wu and He, 2018; Wu and Johnson, 2021). Most notably, BatchNorm assumes that batches are a good approximation of the training data and only performs well when batches are large enough and sampled i.i.d. (Ioffe and Szegedy, 2015; Wu and Johnson, 2021).

This generally does not hold for sequence learning, where batches contain highly correlated, sequential samples, and has led to the use of alternatives such as LayerNorm (LN) (Ba et al., 2016) in NLP (Shen et al., 2020; Vaswani et al., 2017). In video learning, BN has been studied less (Cai et al., 2021; Wu and He, 2018), despite the use of BN-based CNNs.

In natural-video tasks, CNNs are only used to extract image- or clip-wise features using pretrained CNNs (e.g. Carreira and Zisserman

(2017)) off-the-shelf. Only the temporal model, which typically does not contain BN, is trained to aggregate features over time (Abu Farha et al., 2020; Farha and Gall, 2019; Huang et al., 2020; Ishikawa et al., 2021; Ke et al., 2019; Sener et al., 2020; Wang et al., 2020; Yi et al., 2021). However, in specialized small-data domains such as surgical video, well-pretrained CNNs may not be available (Czempiel et al., 2022; Zhang et al., 2022), requiring CNNs to be finetuned, either through 2-stage (Czempiel et al., 2020) or end-to-end (E2E) training (Jin et al., 2017). The latter seems preferable to enable joint learning of visual and temporal features, especially since spatio-temporal feature extractors (e.g. 3D CNNs) have not been effective on small-scale surgical datasets (Czempiel et al., 2022; Zhang et al., 2022). However, BN layers in CNNs pose obstacles for end-to-end learning.

We hypothesize that BN's problems with correlated, sequential samples have silently caused research in video-based surgical workflow analysis (SWA) to head into a sub-optimal direction. The focus has shifted towards developing sophisticated temporal models to operate on extracted image features similar to the natural-video domain, replacing end-to-end learning with complex multi-stage training procedures,

---

* Correspondence to: Department of Translational Surgical Oncology, National Center for Tumor Diseases (NCT/UCC), Fetscherstraße 74, 01307 Dresden, Germany.
E-mail address: dominik.rivoir@nct-dresden.de (D. Rivoir).

where each component (CNN, LSTM Hochreiter and Schmidhuber, 1997, TCN Farha and Gall, 2019, Transformer Vaswani et al., 2017, etc.) is trained individually (Bano et al., 2020; Gao et al., 2021; Kannan et al., 2019; Marafioti et al., 2021; Yuan et al., 2021). We argue that even simple CNN–LSTM models can often outperform these methods when BN-free backbones are used and the model is trained end to end.

We investigate BatchNorm's pitfalls in end-to-end learning on two surgical workflow analysis (SWA) tasks: *surgical phase recognition* (Twinanda et al., 2016) and *instrument anticipation* (Rivoir et al., 2020). We choose these for two reasons: The lack of well-pretrained feature extractors and ineffectiveness of 3D CNNs signify the need for end-to-end approaches in SWA and thus make BN-issues most relevant here. Further, SWA is one of the most active research areas for *online* video understanding, where models are constrained to access only past frames, causing additional BN problems regarding leakage of future information.

Our contributions can be summarized as follows:

1. We challenge the predominance of multi-stage approaches in surgical workflow analysis (SWA) and show that awareness of BN's pitfalls is crucial for effective end-to-end learning.
2. We provide a detailed literature review showing BN's impact on existing training strategies in surgical workflow analysis.
3. We analyze when BN issues occur, including problems specific to online tasks such as "cheating" in anticipation.
4. Leveraging these insights, we show that even simple, end-to-end CNN–LSTM models can be highly effective when BN is avoided and strategies which maximize temporal context are employed.
5. These CNN–LSTMs beat the state of the art on three surgical workflow benchmarks.
6. We reproduce our findings on natural-video datasets to show our study's potential for wider impact.

2. Background & related work

2.1. Batch normalization and beyond

BatchNorm (Ioffe and Szegedy, 2015) is used in almost all modern CNNs. It is assumed to have a regularizing effect, improve convergence speed and enable training of deeper networks (Bjorck et al., 2018; De and Smith, 2020; Hoffer et al., 2017). In BatchNorm, features x are standardized by estimating the channel-wise mean and variance and then transformed using an affine function with learnable parameters γ, β. Formally,

$$BN(x) = \gamma \frac{x - E(x)}{\sqrt{Var(x) + \epsilon}} + \beta, \qquad (1)$$

where $E(x)$ and $Var(x)$ are the mean and variance estimates calculated from the current batch during training. During inference, when the concept of batches is generally not applicable anymore, a running average of the training values of $E(x)$ and $Var(x)$ is used to estimate the global dataset statistics.

Despite BN's popularity and success, it has major disadvantages. The discrepancy between train and test behavior can lead to drops in performance when batch statistics are poor estimates of global statistics, e.g. when batches are small or contain correlated samples. Also, BN's property of depending on other samples in the batch can lead to hidden mistakes if not handled with care. Resulting problems have been observed in many areas such as object detection (Wu and He, 2018; Wu and Johnson, 2021), sequence modeling (Ba et al., 2016), reinforcement learning (Salimans and Kingma, 2016), continual learning (Pham et al., 2022), federated learning (Andreux et al., 2020) and contrastive learning (Henaff, 2020). E.g. in language modeling, both aforementioned issues are highly problematic and have led to the use of alternative normalization layers (Ba et al., 2016; Shen et al., 2020; Vaswani et al., 2017).

Cross-Iteration BatchNorm (CBN) (Yao et al., 2021b), EvalNorm (Singh and Shrivastava, 2019) and others (Ioffe, 2017; Yan et al., 2020; Yang et al., 2021) are extensions of BN which address some of these issues. Normalization layers which only operate on single samples (e.g. GroupNorm Wu and He (2018), LayerNorm Ba et al. (2016)) have also been proposed (Ba et al., 2016; Labatie et al., 2021; Liu et al., 2020; Singh and Krishnan, 2020; Ulyanov et al., 2016; Wu and He, 2018). Dynamic Normalization and Relay (DNR) (Cai et al., 2021) proposed recurrent estimation of affine parameters for video models but does not address the discrepancy of batch and global statistics.

Wu and Johnson (2021) provide the most extensive review of BN issues for vision to date, including train–test discrepancies, the effect of batch sizes and correlated samples as well as solutions such as FrozenBN (Wu and Johnson, 2021) or redefining over which samples to normalize. Related to our work, they show problems of small batches for estimating batch statistics and discuss cheating behavior in object detection and contrastive learning. Consequences for video are only hypothesized. Further, Wu and He (2018) show how BN hides benefits of higher frame rates for clip-based action classification.

Recently, there has been a trend towards CNNs without BN. Brock et al. (2021) discuss disadvantages of BN and propose a class of normalizer-free networks *NFNet*. They, however, require custom optimizers and are sensitive to hyperparameters. Finally, *ConvNeXt* (Liu et al., 2022b) is a CNN competitive with Vision Transformers (Dosovitskiy et al., 2020). It replaces BatchNorm with LayerNorm although this was not the focus of the paper.

2.2. Online surgical workflow analysis

Surgical Workflow Analysis (SWA) (Maier-Hein et al., 2017) is a broad set of tasks which aim at understanding events and their interrelations in recorded or live surgical videos (Fig. 1). SWA covers a wide range of common computer vision problems including temporal action segmentation (e.g. surgical phase recognition Twinanda et al., 2016), dense frame-wise regression (e.g. anticipating instrument usage Rivoir et al., 2020 or procedure duration Twinanda et al., 2018) or video classification (e.g. early surgery type recognition Kannan et al., 2019). SWA tasks are inherently video-based as they require understanding the sequence of events which have occurred until the current time point and often require memorizing long-range dependencies. The overall goal is to provide live, context-aware assistance during surgery (Maier-Hein et al., 2022). Thus, tasks are mostly defined as online recognition problems (Czempiel et al., 2020; Kannan et al., 2019; Rivoir et al., 2020; Twinanda et al., 2018).

Most approaches to surgical workflow tasks combine a 2D CNN for frame-wise feature extraction with a temporal model (e.g. LSTM, TCN or Attention) for aggregation over time. In the following, we discuss different learning strategies which have emerged for surgical workflow tasks and how they were possibly influenced by BN-related problems. Table 1 provides an overview of our findings.

E2E without BN: Early methods completely avoid BN-issues by using backbones without BatchNorm. End-to-end CNN–LSTM models with AlexNet (Krizhevsky et al., 2012) or VGG (Simonyan and Zisserman, 2014) backbones have been proposed for phase recognition (Bodenstedt et al., 2019a; Yengera et al., 2018), anticipation (Rivoir et al., 2020), duration prediction (Rivoir et al., 2019; Yengera et al., 2018), surgery type prediction (Kannan et al., 2019) and occlusion detection (Bano et al., 2020). All methods train long sequences of several minutes. However, results are not competitive due to the use of outdated backbones.

E2E with BN: SV-RCNet (Jin et al., 2017) and MTRCNet (Jin et al., 2020) are two examples of end-to-end ResNet-LSTM models for phase recognition. However, to minimize the discrepancy between batch and global statistics, training sequences are only 10 frames long (2 and 10 s, respectively) and processed in large batches of 100 sequences. TMRNet (Jin et al., 2021) and LRTD (Shi et al., 2020) extend SV-RCNet

Table 1
Previous approaches for video-based surgical workflow tasks. End-to-end approaches on short sequences have been proposed initially. Recent research has favored multi-stage approaches where each model component is trained separately. (($\checkmark$) = contains an end-to-end stage).

| Methods	BN	End2End	Train stages	Remarks
AL-DBN (Bodenstedt et al., 2019a), Tool-Ant (Rivoir et al., 2020)	✗	✓	1	End-to-end CNN–LSTM ✗ *non-SOTA backbone (AlexNet)*
SV-RCNet (Dosovitskiy et al., 2020), MTRCNet (Jin et al., 2020)	✓	(✓)	2	CNN–LSTM w/ end-to-end stage ✗ *many short sequences per batch*
TMRNet (Jin et al., 2021), LRTD (Shi et al., 2020)	✓	(✓)	2	SV-RCNet + Attention ✗ *many short sequences per batch*
CataNet (Marafioti et al., 2021)	✓	(✓)	4	CNN–LSTM w/ end-to-end stage ✗ *requires 4 training stages*
TransSVNet (Gao et al., 2021), TeCNO (Czempiel et al., 2020) and others	✓	✗	2–3	CNN+{RNN,TCN,Transformer} ✗ *separate training stages for each model component*
Proposed	✗	✓	1	End-to-end CNN–LSTM *SOTA using BN-free backbones*
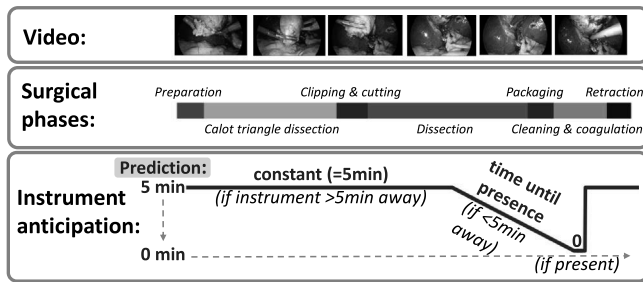
Fig. 1. Two surgical workflow tasks on the Cholec80 dataset (Twinanda et al., 2016): *phase recognition,* a special case of temporal action segmentation, and *instrument anticipation,* defined as predicting the time until occurrence of an instrument within a specified horizon.

with attention blocks to learn long-range dependencies. Still however, sequences of only 10 frames are trained end to end. CataNet (Marafioti et al., 2021) proposes a complex, 4-stage learning process for ResNet-LSTMs to predict surgery duration. In one of the stages, end-to-end learning on long sequences is made possible by using FrozenBN with global statistics from a previous stage.

Multi-stage: Recently, methods have shifted towards more complex temporal models on top of pretrained visual features. BN-related issues are circumvented by first training a CNN on randomly sampled image batches, followed by the temporal model trained on frozen image features. Methods in this style have been proposed for phase recognition (Czempiel et al., 2020, 2021; Gao et al., 2021; Yi et al., 2022; Zisimopoulos et al., 2018), duration prediction (Aksamentov et al., 2017; Bodenstedt et al., 2019b; Twinanda et al., 2018), tracking (Nwoye et al., 2019) or anticipation (Yuan et al., 2021). Most notably, TeCNO (Czempiel et al., 2020), an MS-TCN (Farha and Gall, 2019) trained on ResNet features, is the popular approach for 2-stage learning and Trans-SVNet (Gao et al., 2021), a 3-stage method which trains a Transformer on TeCNO features, is often considered state of the art among methods with CNN backbones. More recently, few multi-stage methods have achieved strong performance by using Transformer-based backbones (Chen et al., 2022; He et al., 2022; Liu et al., 2023a,b).

Awareness for BN issues: Although BatchNorm has possibly influenced the move towards complex, multi-stage or short-range models, its issues have rarely been discussed. In their end-to-end model, Yengera et al. (2018) justify the use of the BN-free AlexNet with efficiency reasons. Kannan et al. (2019) simply state that their end-to-end model with VGG outperformed ResNet, but not whether this may be due the absence to BatchNorm. The authors of SV-RCNet (Jin et al., 2017)

do not explain their choice of short sequences despite large batch sizes. Nwoye et al. (2019) justify their 2-stage approach through "fair comparison". CataNet (Marafioti et al., 2021) uses FrozenBN in an end-to-end learning stage,[1] but this is not mentioned in the paper. Rivoir et al. (2020) briefly mention a cheating phenomenon to justify the choice for an AlexNet in instrument anticipation.

3D backbones: It is noticeable that surgical workflow methods almost unanimously use 2D backbones as feature extractors. The specialized surgical domain requires backbones to be finetuned and several studies suggest that the small-scale public datasets available in this domain are not sufficient to train large 3D CNNs (Czempiel et al., 2022; Zhang et al., 2022). Only few works achieve good results on larger private datasets (Czempiel et al., 2022; Zhang et al., 2021). More recently, He et al. (2022) confirmed poor performance of 3D CNNs but showed promising results using the spatio-temporal Transformer architecture *Video Swin* (Liu et al., 2022c) on short 1-s sequences.

2.3. Action segmentation & anticipation

In the natural-video domain, action segmentation and anticipation can be viewed as workflow analysis tasks as they aim at understanding the composition of complex activities.

Training strategies: Methods avoid BatchNorm during temporal learning by almost exclusively relying on pretrained visual features (Carreira and Zisserman, 2017) in both action segmentation (Farha and Gall, 2019; Huang et al., 2020; Ishikawa et al., 2021; Wang et al., 2020; Yi et al., 2021) and anticipation (Abu Farha et al., 2020; Furnari and Farinella, 2020; Gao et al., 2017; Ke et al., 2019; Sener et al., 2020). Whether this development can be attributed to computational costs, the availability of well-pretrained feature extractors or BatchNorm's properties is not yet clear.

Sener et al. (2020) indicate limitations of pretrained visual features for action anticipation by showing a large performance gap compared to using ground-truth observed actions as input. The only end-to-end anticipation model for natural video of which we are aware (AVT, Girdhar and Grauman (2021)) does not use BatchNorm as it is entirely Transformer-based. Although not mentioned as motivation for this model choice, BN would likely have caused both train–test discrepancy and "cheating" since single-sequence batches are used. Recently, Liu et al. (2022a) show the effectiveness of end-to-end strategies for the related temporal action detection task. They use FrozenBN but do not discuss BatchNorm issues.

Online recognition: Online action understanding has mostly focused on temporal action detection (Eun et al., 2020; Lea et al., 2017;

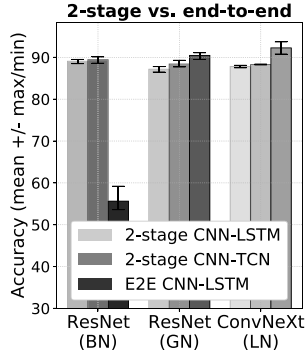[1] https://github.com/aimi-lab/catanet/blob/main/train_rnn.py#L179.

Fig. 2. *Phase Recognition (Cholec80):* BN-free, end-to-end CNN–LSTMs outperform 2-stage LSTMs and TCNs. The widely used ResNet50 with BN fails. Split: 40/8/32 from TeCNO (Czempiel et al., 2020).

Xu et al., 2019; Zhao et al., 2020) or anticipation. Apart from few exceptions (Ghoddoosian et al., 2022), action segmentation is usually done offline (Farha and Gall, 2019; Huang et al., 2020; Ishikawa et al., 2021; Wang et al., 2020; Yi et al., 2021).

3. Making end-to-end video learning work

We analyze limitations and impact of BatchNorm for end-to-end learning on two surgical tasks (Fig. 1). In *surgical phase recognition*, an online temporal segmentation task, we show how BatchNorm can affect training strategies and performance. Especially, we show that supposedly outdated CNN–LSTMs can be very effective when BN's pitfalls are taken into consideration. Then, we demonstrate BatchNorm's ability to access future frames in online tasks through a "cheating" phenomenon in *surgical instrument anticipation*.

3.1. Proposed end-to-end learning strategies

Our premise is that end-to-end learning is preferable over multi-stage approaches and thus we propose a simple, intuitive strategy for training CNN–LSTM models on online surgical workflow tasks. We use the LSTM to demonstrate that even simple temporal aggregation can be effective when visual features are improved through end-to-end learning and temporal context is maximized. We also argue that recurrent, state-based models, where a memory state propagates information through time, are often well suited for online recognition tasks (Furnari and Farinella, 2020; Gao et al., 2017; Ghoddoosian et al., 2022). Especially for end-to-end learning on partial video sequences, the LSTM's hidden state can be utilized to increase temporal context. Specifically, we propose:

End-to-end training: We optimize the feature backbone and temporal model end to end in a single training stage.

Single-sequence batch: In end-to-end settings, batches are limited to partial clips containing only few minutes of total video due to memory constraints. We maximize temporal context by using a single sequence/clip of 64–256 frames per batch.

Carry-hidden training (CHT) *(optional)*: We select batches in temporal order. This way, the LSTM's detached hidden state can be carried across batches to further increase temporal context during training.

Partial freezing *(optional)*: We freeze bottom layers of the backbone to increase the length of training sequences. Although this is not fully end to end anymore, the main benefits are still given: There is still only a single training stage and image features are optimized in a temporal context.

End-to-end learning (Jin et al., 2016, 2017), single-sequence batches (Kannan et al., 2019) and CHT (Nwoye et al., 2019) have been used individually in BN-based approaches. However, only methods with

BN-free backbones such as AlexNet have been able to employ these (Bodenstedt et al., 2019a) or similar (Yengera et al., 2018) strategies in combination. This is because end-to-end training is not compatible with single-sequence batches if the backbone contains BatchNorm, and by extension with CHT as it requires single-sequence batches. The following paragraph elaborates the main problems.

3.2. BatchNorm's problems with sequential data

(1) Small-batch effect (Section 4.2, Figs. 2, 3): It is well-known that BatchNorm performs poorly with small batches (Wu and He, 2018; Wu and Johnson, 2021). Batch statistics, used to normalize features during training, only poorly estimate global statistics used during inference and thus induce a discrepancy between train and test behavior. However, when batches contain sequential video frames, even large batches lead to similar issues as the highly correlated samples also do not approximate global statistics well. Similar observations have been made in different contexts (Ba et al., 2016; Wu and He, 2018).

(2) Cheating/leakage (Section 4.3, Figs. 6, 7): Previous work has discussed BatchNorm's ability to leak information from other samples in the batch and its impact on performance in object detection or contrastive learning (Wu and Johnson, 2021). In online video tasks, a similar phenomenon can be fatal. BatchNorm can leak information from future frames, which allows models to "cheat" certain objectives and prevents learning useful features. This "cheating" phenomenon is most obvious in anticipation tasks. In *instrument anticipation*, the occurrence of an instrument influences the batch statistics of channels reacting to its appearance. During training, BN models may pick up on these fluctuations in batch statistics even at *earlier time points* within the same batch and use them to anticipate the instrument's later occurrence. Thus, the model only has to learn to *recognize* the instrument to solve the *anticipation* objective during training, but will fail during inference when batch statistics are not available.

3.3. Alternative backbones without BatchNorm

We enable end-to-end learning with single-sequence batches by replacing ResNet50 with existing BN-free CNNs. We use a ResNet50 with **GroupNorm (GN)** (Wu and He, 2018) for fair comparison as well as the more recent **ConvNeXt-T** (Liu et al., 2022b), which uses ViT-style LayerNorm (Yao et al., 2021a) and is of comparable size to ResNet50. We also investigate **FrozenBN** (Liu et al., 2022a; Wu and Johnson, 2021), where batch statistics are replaced by global statistics from the pretraining stage. All models are pretrained on ImageNet-1k (Krizhevsky et al., 2012).

3.4. Limitations of previous strategies

Existing BatchNorm-based approaches avoid issues with sequential data either through multi-stage training or multi-sequence batches. Both solutions, however, suffer from potential limitations which we attempt to explain in the following. We support these claims through our experiments.

Multi-stage: Multi-stage training is the most popular approach and effectively avoids BN's issues by training backbones without temporal context. Yet, this has several disadvantages. Firstly, it increases the number of hyperparameters since learning rate, number of epochs etc. have to be tuned for each training stage. Secondly, when backbone and temporal model are trained on the same data, overfitting on the image task can prevent the temporal model from learning useful features. For example, we found early stopping of the ResNet pretraining crucial to reproduce TeCNO's (Czempiel et al., 2020) results. Lastly and most importantly, pretraining on single frames changes the dynamics of the task and cannot guarantee features that are suitable for the target task. A certain visual feature might only be relevant to the task in combination with a previous event. If the backbone does not learn
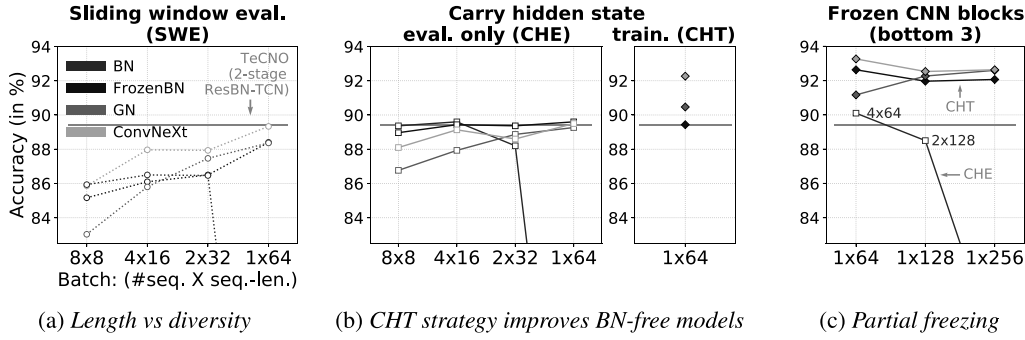
(a) *Length vs diversity*   (b) *CHT strategy improves BN-free models*   (c) *Partial freezing*

Fig. 3. *Phase Recognition (Cholec80):* (a) While longer sequences improve BN-free models, BN in ResNet50 backbones requires careful selection of the number of sequences per batch before performance drops. (b) Carrying hidden state through entire videos at test time improves models. By also doing this during training, train–test discrepancy is removed and models improve further. In BN models, implementing this is not straightforward. (c) BN models improve by freezing backbone blocks and thus increasing training sequences while maintaining batch diversity. Yet, freezing also improves BN-free models. Explicit scores with standard deviations and additional metrics can be found in the appendix (Tables A.7 & A.8).
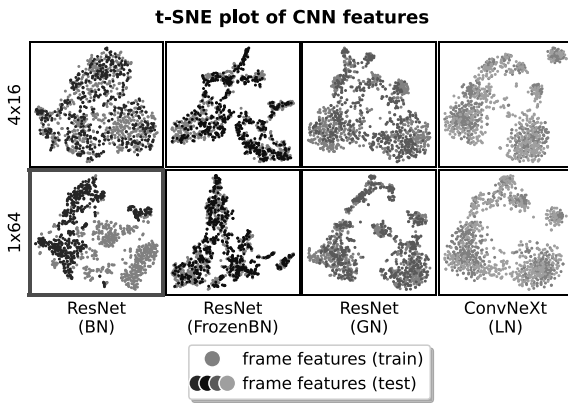

Fig. 4. *Phase recognition (Cholec80):* With multi-sequence batches 4 × 16, train and test features cover similar distributions. In BN-based backbones, single-sequence batches (1 × 64) cause a shift in train and test features due to the discrepancy between batch and global statistics.

to capture this feature during pretraining, this information is lost and cannot be recovered in the subsequent temporal-modeling stage.

End-to-end with multi-sequence batches: Sampling multiple sequences per batch (e.g. SV-RCNet Jin et al., 2017) reduces the discrepancy between batch and global statistics. However, this comes at the cost of shorter training sequences. It also restricts possible training strategies like carrying hidden states of LSTMs across batches and can thus impair performance.

4. Experiments

We provide a comprehensive and detailed analysis of how Batch-Norm affects end-to-end surgical workflow analysis. We show the advantage of end-to-end over 2-stage learning *(Hypothesis 1)*, longer training sequences *(H.2)* and carrying hidden states across batches in online tasks *(H.3)* and how this can fail using BN-based backbones. We confirm our findings on similar, natural-video datasets *(H.4)* and other BN variants *(H.5)*. Then, we discuss how freezing layers can alleviate BN-issues *(H.6)* and position our proposed models within the state of the art for surgical phase recognition *(H.7)*. We show BN's potential to cheat *(H.8)*, provide evidence in instrument anticipation *(H.9)* and compare models to previous anticipation approaches *(H.10)*.

4.1. Data, tasks & implementation details

Cholec80: For our main analysis, we train phase recognition and anticipation models on the widely-studied *Cholec80 dataset* (Twinanda

et al., 2016), which consists of 80 recorded gallbladder removals (cholecystectomies). Videos range from 12 to 100 min (mean ca. 38 min), processed at 1fps with frame-wise labels for 7 instruments and 7 surgical phases.

Surgical phase recognition is an dense online temporal action segmentation task (Twinanda et al., 2016). We use the mean video-wise accuracy for evaluation (F1 in Appendix Table A.8).

Instrument anticipation is an online frame-wise regression task. At each time point, the objective is to predict the remaining time until occurrence of an instrument within a horizon of 5 min. Outside the horizon, a constant should be predicted (see Fig. 1). We follow the recently proposed task formulation (Rivoir et al., 2020). We use the weighted MAE (wMAE) as evaluation metric, where errors inside and outside the horizon are weighted equally to counteract imbalances.

For both tasks, we use 40 videos for training, 8 for validation and 32 for testing, following TeCNO (Czempiel et al., 2020). Epochs with the best validation score (*acc./wMAE*) are selected for testing. For SOTA comparison in phase recognition, we retrain models on a 32/8/40 split. This allows us to stay comparable to the widely used 40/40 split (Twinanda et al., 2016; Gao et al., 2021) while still having a validation set for model selection. All runs are repeated 3 times. More training and evaluation details can be found in Appendices E & F.

GTEA & 50Salads: To reproduce our claims on natural videos, we retrain phase recognition models for online action segmentation on *GTEA* (Fathi et al., 2011) and *50Salads* (Stein and McKenna, 2013).

AutoLaparo & CATARACTS: For state-of-the-art comparisons, we retrain phase recognition models on two additional surgical datasets. AutoLaparo (Wang et al., 2022) contains 21 long videos (10/4/7 split) ranging from 27 to 112 min (mean ca. 66 min) with 7 coarse phases annotated. CATARACTS (Zisimopoulos et al., 2018) contains 50 short videos ranging from 6 to 40 min (mean ca. 11 min) with 19 fine-grained steps annotated. To be able to compare to the 25/25 split used in previous work (Chen et al., 2022), we use a 20/5/25 split. On both datasets, we report the mean video-based accuracy as well as macro-precision, -recall and -jaccard, which are computed over all test frames. See Appendices E & F for details.

4.2. Surgical phase recognition

Hypothesis 1: End-to-end outperforms 2-stage training, but not with BatchNorm. We argue that end-to-end learning is preferable over 2-stage approaches but fails when the backbone contains BatchNorm. To demonstrate this, we train CNN–LSTM models with three different backbones (Resnet50-BN, ResNet50-GN and ConvNeXt-T), single-sequence batches of 64 frames (1 × 64) and the proposed carry-hidden training (CHT). For comparison, we train each backbone in a 2-stage process with an LSTM or causal MS-TCN, following TeCNO's training procedure (Czempiel et al., 2020).

Fig. 2 shows that end-to-end models outperform 2-stage approaches when using a backbone without BN. Only the typically used BN-based ResNet50 fails completely. While BatchNorm is known to fail with small batch sizes (Wu and He, 2018; Wu and Johnson, 2021), even large batch sizes lead to similar issues here due to the strong correlation of subsequent frames. In other words, a batch with a single 64-frame sequence behaves similar to a batch size of 1. The next hypothesis investigates this in more detail.

Hypothesis 2: BatchNorm causes trade-off between sequence length and batch diversity. We investigate the effect of different batch configurations and follow SV-RCNet's (Jin et al., 2017) strategy for end-to-end CNN–LSTMs: Multiple shorter sequences are sampled per batch, hidden states are reset after each training sequence and the LSTM uses a sliding window for evaluation *(SWE)* to ensure consistent sequence lengths at train and test time. We test this strategy using 1–8 sequences in batches of 64 frames in total.

In Fig. 3(a), we observe that the performance of all BN-free models improves with fewer but longer sequences per batch since it increases the temporal context. In BN models, however, fewer sequences poorly approximate global statistics, leading to a discrepancy in feature normalization at train and test time. Thus, we observe drops in performance despite the constant total batch size, similar to the frame-rate trade-off in video-clip classification (Wu and He, 2018). The BN backbone outperforms its FrozenBN and GroupNorm counterparts for multiple short sequences but stagnates and then collapses with fewer, longer ones. Due to this trade-off between sequence length and batch diversity, BN models eventually cannot reach the performance of the fully-sequential, BN-free models.

To show BatchNorm's train–test discrepancy, we visualize CNN features in Fig. 4 with t-SNE plots (perplexity: 30). When training with multiple short sequences per batch 4×16, train and test features cover very similar distributions for all backbones. With single-sequence batches 1×64, however, we observe a shift between train and test features for BN-based backbones, while this issue does not occur with BN-free backbones.

Hypothesis 3: BatchNorm restricts LSTM usage for online tasks. Note that by adopting SV-RCNet's hidden-state handling, none of the methods outperform the 2-stage SOTA (Czempiel et al., 2020). This is likely due to the short temporal context seen by the LSTM. Simply carrying the hidden state across the entire video during inference *(CHE)* already provides a clear performance boost without changing the training strategy (Fig. 3, *SWE vs. CHE*). Yet, this induces another train–test discrepancy since the LSTM accumulates hidden states over much longer time periods at test time compared to training time, which could cause unpredictable behavior.

In online tasks, however, the discrepancy can naturally be resolved by exploiting the fact that hidden states only flow in a single direction. Specifically, we can implement the proposed *carry-hidden training (CHT)* by selecting batches in temporal order and carrying the detached hidden states across batches during training as well.

With this approach, temporal context is increased during training as well as inference. Now, GN and ConvNeXt backbones outperform 2-stage and other end-to-end strategies (Fig. 3(b)). In BN models, however, this learning strategy is not implementable in a straightforward way since it requires selecting batches in temporal order, which violates the i.i.d. requirement. Thus, the train–test discrepancy in LSTMs remains.

Interestingly, the *CHT* strategy is not effective with FrozenBN. We found models often to collapse and finally only achieve subpar performance by lowering the initial learning rate. The lack of proper normalization with FrozenBN might cause more instability than in GN or ConvNeXt models and we suspect that the non-i.i.d. selection of batches during *CHT* can encourage this. We present more evidence in our experiments on other datasets (*Hypothesis 4*, Fig. 5) and a more detailed exploration of learning rates in the appendix (Table A.9).
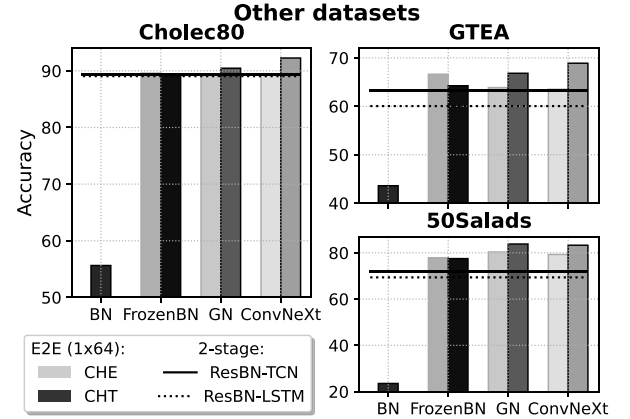

Fig. 5. *Online action segmentation:* Results on surgical data (left) can be reproduced on natural-video datasets (right).

Hypothesis 4: BatchNorm induces similar issues on natural-video datasets. Fig. 5 confirms our findings on small-scale datasets from the natural-video domain (*GTEA* and *50Salads*) by retraining the same models used for phase recognition on online action segmentation. We find that our main claims can be reproduced, namely that (1) BN-based end-to-end models fail with single-sequence batches, (2) end-to-end approaches without BN outperform 2-stage models and (3) the CHT strategy is effective using GN or ConvNeXt but not with FrozenBN. Our proposed strategies aim at increasing temporal context, and thus seem mostly applicable to tasks where this is relevant (e.g. long videos and workflow-based tasks such as cooking or other complex activities). Its impact on other video tasks remains open.

Hypothesis 5: Other BatchNorm variants are not suitable. In Table 2, we perform a more extensive comparison of batch-independent and batch-dependent normalization layers. We find that other batch-independent methods (InstanceNorm, LayerNorm) benefit from our context-enlarging strategies. However, even BatchNorm variants which attempt to alleviate issues with temporal dependence or small batches exhibit similar failure cases as BatchNorm. Temporal dependence can be avoided entirely by normalizing each time step individually. Yet, with single-sequence batches this still fails due to normalization over single frames. DNR (Cai et al., 2021) proposes temporally dependent estimation of BN's affine parameters to adjust to properties of batches in video settings. Yet, batch stats are still computed like in standard BN, so the train/test discrepancy with single-sequence batches remains and model collapse with 1×64 batches can clearly be observed. CBN (Yao et al., 2021b) reduces batch dependence by accumulating stats from multiple batches. Yet, for our proposed CHT strategy, CBN is not suitable due to strong inter-batch correlations of subsequent batches. Further, we found the transition to accumulated stats after its burn-in phase to cause instability even in CHE settings, despite finetuning. Expectedly, this issue was stronger in the single-sequence case (1×64, CHE).

Hypothesis 6: Freezing backbone layers can improve BN models, but also their BN-free counterparts. As argued before, the trade-off between sequence length and batch diversity is a disadvantage of BN-based models. By freezing parts of the CNN backbones, sequence lengths can be increased while maintaining the same number of sequences. Hence, we freeze the bottom three blocks of all backbones and retrain each model with sequence lengths of 64, 128 and 256. For BN-free models, we choose the CHT strategy but resort to CHE for the BN model (Fig. 3(c)).

Through freezing, 4×64-BN finally outperforms the 2-stage SOTA but collapses again with longer sequences while GN models improve. Even FrozenBN matches other BN-free models. Consistent with prior beliefs (He et al., 2016), lack of normalization might be less problematic in shallow models.

Table 2
Phase recognition (Cholec80): Comparison of batch-independent and -dependent normalization techniques.

Accuracy (mean ± std. over runs) on 40/8/32 split from TeCNO (Czempiel et al., 2020).			
End-to-end ResNet-LSTM w/ ...	4 × 16, CHE	1 × 64, CHE	1 × 64, CHT
Batch-**independent** normalization			
InstanceNorm (Ulyanov et al., 2016)	87.3(±2.5)	88.5(±0.5)	**90.2(±0.2)**
LayerNorm (Ba et al., 2016)	87.6(±1.6)	89.1(±1.3)	**90.4(±2.8)**
GroupNorm (Wu and He, 2018)	87.9(±0.8)	89.3(±0.4)	**90.5(±0.9)**
Batch-**dependent** normalization			
BatchNorm w/ norm. over dim. B,T,H,W (standard)	89.6(±1.0)	59.7(±3.8)	55.6(±3.1)
BatchNorm w/o norm. over temp. dim. T (only B,H,W)	88.4(±0.5)	62.7(±7.1)	44.6(±0.7)
DNR (Cai et al., 2021)	88.9(±0.7)	68.0(±6.5)	61.6(±2.7)
CBN (Yao et al., 2021b) w/ window size 1 *(equiv. to BN)*	89.5(±0.7)		
CBN w/ burn-in 3 epochs, window size 8 *(original)*	49.1(±2.4)		
CBN w/ burn-in 10 epochs, window size 4	87.4(±0.8)	79.4(±1.3)	51.5(±19.8)

Hypothesis 7: Simple end-to-end CNN–LSTMs achieve state-of-the-art performance. Finally, we compare our proposed approaches (end-to-end and partial freezing) to the state of the art in phase and step recognition on three datasets.[2]

Cholec80: To be comparable to previous work (Gao et al., 2021; Jin et al., 2017, 2021), we retrain our models on a 32/8/40 split. As evaluation metrics, we report the mean video-wise accuracy and the mean video-wise balanced accuracy (i.e. mean video-wise macro recall) as well as variants of accuracy and Jaccard scores with relaxed boundaries, i.e. where errors at phase transitions are ignored under certain conditions. Relaxed metrics were proposed in the *M2CAI 2016 Challenge* (Stauder et al., 2016; Twinanda et al., 2016). Table 3 shows our results.

Through partial freezing on 256-frame sequences, all our models with BN-free backbones (FrozenBN, GN and ConvNeXt) outperform the previous state of the art, including complex multi-stage methods with attention-based temporal models (Gao et al., 2021; Jin et al., 2021; Chen et al., 2022), complex augmentation pipelines (Yi et al., 2022) or Transformer backbones (Chen et al., 2022; He et al., 2022). Models trained completely end to end (E2E) on shorter 64-frame sequences perform slightly worse. However, the ConvNeXt variant still achieves SOTA performance and the GN variant beats or matches other ResNet-based methods. Note that Not-E2E (Yi et al., 2022) achieves slightly higher relaxed accuracy but much lower relaxed Jaccard. The completely end-to-end FrozenBN model underperforms, as already observed in *Hypothesis 3*.

While the newer ConvNeXt backbone is not necessarily comparable to ResNet backbones, note that our proposed strategies are necessary to achieve strong performance. Specifically, ConvNeXt models do *not* perform better than its ResNet variants with the *CHE* strategy (Fig. 3(b)) or in 2-stage settings (Fig. 2 & Table A.7). These results further indicate that improving visual-feature learning in simple end-to-end approaches might be more effective than designing more complex temporal models on sub-optimal visual features.

Finally, BN-based end-to-end models perform poorly with SV-RCNet's sliding-window evaluation (SWE) but improve by simply carrying hidden states during inference (CHE) without changing the training procedure. This indicates that previous end-to-end approaches were evaluated poorly. By freezing backbone layers, BN-based CNN–LSTMs also achieve competitive results and match the 64-frame, end-to-end GN model. Performance of ConvNeXt or 256-frame GN and FrozenBN models is however not reached. This shows that, with careful design, BN can still be effective. Still, models without BatchNorm enable

simpler and a wider range of training procedures with longer training sequences and ultimately perform better.

AutoLaparo: With AutoLaparo containing long videos like Cholec80, we can reproduce our main findings (Table 4): Maximizing temporal context through the proposed strategies (Partial freezing, 1 × 256, CHT) outperforms its ablations without single-sequence batches and CHT (Partial freezing, 4 × 64, CHE) or partial freezing (E2E 1 × 64, CHT). These models even outperform the state of the art by a margin, including very recent, fully transformer-based approaches LoViT and SKiT. Although BatchNorm models perform very well with partial freezing, employing single-sequence batches and CHT, which works best with all other backbones, expectedly fails with BatchNorm.

CATARACTS: Since CATARACTS contains shorter videos with more fine-grained annotations, the benefits of our context-maximizing strategies are expectedly smaller here. Nevertheless, we can still reproduce several findings (Table 5): Single-sequence CHT (E2E, 1 × 64, CHT) outperforms multi-sequence batching (E2E, 4 × 16, CHE) with BN-free backbones but fails with BatchNorm. Partial freezing did not have a clear benefit and even underperformed with GN — likely due to fewer long-term dependencies. Despite not reaching SOTA performance (as the fully Transformer-based model by Chen et al. (2022) performs exceptionally well), our simple ConvNeXt and GN models consistently achieve the second highest scores, outperforming Transformer- and TCN-based multi-stage approaches (Trans-SVNet, TeCNO) as well as 3D-CNNs. BN-based models do not reach this level and again fail when using CHT, which was most effective for BN-free backbones.

4.3. Anticipation of instrument usage

Hypothesis 8: BatchNorm enables "cheating" in certain tasks. BatchNorm can leak information within batches to "cheat" certain objectives. While previous work finds empirical evidence for this effect through degrading performance (Wu and Johnson, 2021), we conclusively demonstrate it by carefully designing a toy task: We propose an impossible task which cannot be solved without "cheating": *Given training batches of size 2, predict for each input sample whether it is equal to the other sample in the batch.* We use a small neural network with BatchNorm (Fig. 6, left) and with single binary inputs $x_i \in \{0, 1\}$. Although this task is clearly unsolvable when each sample in the batch is processed individually, the BN model "cheats" by making predictions based on batch statistics (Fig. 6, right). Models expectedly fail (always predict *False*) when using global statistics during inference or networks without BN.

Hypothesis 9: "Cheating" can cause BN models to fail in instrument anticipation. To visualize how BN can "cheat" during training in anticipation tasks, we evaluate BN in training mode, i.e. processing test videos offline in batches with sequence lengths equal to the training lengths and use batch (instead of global) statistics for normalization.

---

[2] The unconventional classification of SKiT/LoViT as concurrent on Cholec80 and previous work on AutoLaparo is a result of AutoLaparo experiments being added to this paper at a later time point.

Table 3
State of the art in surgical phase recognition on Cholec80 (Twinanda et al., 2016) with a 40/40 split. Our models are trained on a 32/8/40 split to allow for model selection. We find that simple, end-to-end CNN–LSTMs match or outperform multi-stage methods with attention-based temporal models (Gao et al., 2021; Jin et al., 2021; Chen et al., 2022), complex augmentation pipelines (Yi et al., 2022) or Transformer backbones (Chen et al., 2022; He et al., 2022).

Cholec80								
Method	Backbone	Temporal model	#Training stages	Relaxed accuracy (std. over videos)	Relaxed Jaccard (std. over phases)	Accuracy (std. over runs)	Jaccard (std. over runs)	Balanced accuracy (std. over runs)
SV-RCNet (Jin et al., 2017) (E2E, 100 × 10, SWE)	ResNet-BN	LSTM	2	85.3(±7.3)		85.0	65.5	
TeCNO (Czempiel et al., 2020)	ResNet-BN	TCN	2	88.6(±7.8)	75.1(±6.9)	88.1(±0.3)[a]	69.4(±0.5)[a]	85.4(±1.0)[a]
TMRNet (Jin et al., 2021)	ResNet-BN	LSTM, Attn.	2	89.2(±9.4)	78.9(±5.8)			
Trans-SVNet (Gao et al., 2021)	ResNet-BN	TCN, Transf.	3	90.1(±7.1)	79.3(±6.6)	89.0(±1.1)[a]	71.5(±2.4)[a]	85.4(±1.4)[a]
Not-E2E (Yi et al., 2022)	ResNet-BN	TCN, GRU	3	92.0(±5.3)	77.1(±11.5)			
TMRNet (Jin et al., 2021)	ResNeST	LSTM, Attn.	2	90.1(±7.6)	79.1(±5.7)			
PATG (Kadkhodamohammadi et al., 2022)	SE-ResNet	GNN	2			91.4		
Zhang et al. (2022)	C3D	LSTM	2	85.9(±7.9)				
He et al. (2022)	I3D	GRU	2			88.3(±1.0)		
TeSTra (Zhao and Krähenbühl, 2022)	Transf.	Transf.	2			90.1	71.6	
He et al. (2022)	Video-Transf.	GRU	2			90.9(±0.01)		
Dual Pyramid (Chen et al., 2022)	Transf.	Transf.	2			91.4	75.4	
Ours w/ BN								
E2E (4 × 16, SWE) (comparable to SV-RCNet)	ResNet-BN	LSTM	1	87.5(±7.7)	73.1(±8.5)	86.6(±1.1)	66.3(±0.8)	82.9(±0.6)
E2E (4 × 16, CHE)			1	90.3(±7.1)	77.5(±8.5)	89.3(±0.7)	70.4(±1.0)	84.9(±0.8)
Partial freezing (4 × 64, CHE)			1	91.3(±6.9)	**79.8(±9.2)**	90.1(±0.5)	72.8(±1.2)	**86.4(±1.2)**
Ours w/o BN								
E2E (1 × 64, CHT)	ResNet-FrozenBN	LSTM	1	90.0(±7.5)	77.3(±8.5)	89.0(±0.3)	69.4(±1.1)	**85.5(±0.6)**
Partial freezing (1 × 256, CHT)			1	**92.5(±5.9)**	**81.2(±9.6)**	**91.5(±0.5)**	74.2(±1.2)	**88.0(±0.8)**
E2E (1 × 64, CHT)	ResNet-GN	LSTM	1	91.4(±7.8)	78.9(±9.7)	90.3(±1.3)	71.5(±3.0)	**86.0(±1.3)**
Partial freezing (1 × 256, CHT)			1	**93.1(±5.4)**	**80.2(±11.7)**	**92.1(±0.8)**	73.9(±1.9)	**86.4(±0.9)**
E2E (1 × 64, CHT)	ConvNeXt	LSTM	1	**92.4(±6.9)**	**82.0(±7.9)**	91.3(±0.2)	74.6(±0.6)	**87.4(±0.7)**
Partial freezing (1 × 256, CHT)			1	**93.5(±6.5)**	**82.9(±10.1)**	**92.4(±0.3)**	**76.9(±1.0)**	**87.9(±1.3)**
Concurrent work								
LoViT (Liu et al., 2023a)	Transf.	Transf.	2	*92.4 (±6.3)*	*81.2 (±9.1)*	*91.5*	*74.2*	
SKiT (Liu et al., 2023b)	Transf.	Transf.	2	*93.4 (±5.2)*	*82.6*	*92.5*	*76.7*	

[a] Reproduced.

Table 4
State of the art in surgical phase recognition on AutoLaparo (Wang et al., 2022) with 10/4/7 split. We report mean ± std. over runs.

AutoLaparo				
Method	Accuracy	Precision	Recall	Jaccard
SV-RCNet (Jin et al., 2017)	75.6	64.0	59.7	47.2
TMRNet (Jin et al., 2021)	78.2	66.0	61.5	49.6
TeCNO (Czempiel et al., 2020)	77.3	66.9	64.6	50.7
Trans-SVNet (Gao et al., 2021)	78.3	64.2	62.1	50.7
LoViT (Liu et al., 2023a)	81.4	**85.1**	65.9	56.0
SKiT (Liu et al., 2023b)	82.9	81.8	70.1	59.9
ResNet-BN-LSTM				
E2E (4 × 16, CHE)	82.3(±0.5)	71.9(±1.1)	67.8(±0.8)	57.7(±1.2)
Partial freezing (4 × 64, CHE)	**85.2(±2.3)**	78.0(±2.4)	**73.3(±4.7)**	**64.1(±5.7)**
Partial freezing (1 × 256, CHT)	49.4(±12.8)	54.0(±3.7)	42.1(±13.4)	26.7(±10.0)
ResNet-GN-LSTM				
E2E (1 × 64, CHT)	80.3(±1.8)	72.4(±3.0)	65.5(±2.7)	55.6(±3.1)
Partial freezing (4 × 64, CHE)	82.7(±1.3)	73.5(±0.6)	68.1(±2.1)	58.4(±2.4)
Partial freezing (1 × 256, CHT)	**85.7(±0.8)**	73.7(±2.0)	**71.5(±1.3)**	**63.0(±1.3)**
ConvNeXt-LSTM				
E2E (1 × 64, CHT)	**83.0(±1.2)**	73.8(±0.5)	66.8(±1.4)	57.7(±1.4)
Partial freezing (4 × 64, CHE)	82.8(±0.8)	79.3(±4.7)	66.3(±0.8)	57.5(±1.1)
Partial freezing (1 × 256, CHT)	**86.8(±1.5)**	78.2(±4.4)	**72.0(±1.9)**	**64.2(±2.2)**

Table 5

State of the art in surgical step recognition on CATARACTS (Zisimopoulos et al., 2018) with a 25/25 split. Our models are trained on a 20/5/25 split to allow for model selection. State-of-the-art scores are printed bold and we underline scores which outperform all methods except Chen et al. (2022).

| CATARACTS				
Method	Accuracy	Precision	Recall	Jaccard
3D-CNN (Funke et al., 2019)	80.1	66.2	55.7	45.9
SV-RCNet (Jin et al., 2017)	81.3	66.0	57.0	47.2
TeCNO (Czempiel et al., 2020)	79.0	62.6	56.9	45.1
Trans-SVNet (Gao et al., 2021)	77.8	61.3	55.0	43.8
Dual Pyramid (Chen et al., 2022)	**84.2**	69.3	**66.4**	**53.7**
ResNet-BN-LSTM				
E2E (4 × 16, CHE)	79.3(±1.4)	63.0(±2.6)	55.7(±2.4)	44.8(±2.7)
E2E (1 × 64, CHT)	40.4(±5.0)	35.4(±8.8)	20.5(±6.2)	11.8(±4.3)
Partial freezing (4 × 64, CHE)	79.9(±1.7)	60.4(±2.7)	55.9(±0.4)	43.7(±0.9)
ResNet-GN-LSTM				
E2E (4 × 16, CHE)	80.8(±1.4)	67.5(±0.9)	57.3(±2.3)	47.1(±2.9)
E2E (1 × 64, CHT)	82.1(±0.6)	64.0(±2.1)	58.4(±1.1)	47.5(±0.9)
Partial freezing (1 × 256, CHT)	80.4(±2.4)	66.1(±5.4)	58.9(±4.8)	47.2(±5.1)
ConvNeXt-LSTM				
E2E (4 × 16, CHE)	83.0(±1.2)	**69.9(±2.3)**	59.7(±1.1)	49.3(±0.6)
E2E (1 × 64, CHT)	83.8(±0.8)	63.7(±5.7)	60.9(±2.7)	49.7(±3.6)
Partial freezing (1 × 256, CHT)	83.3(±0.5)	66.8(±3.5)	61.8(±1.7)	50.3(±0.5)
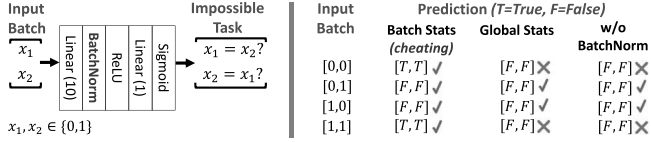
Fig. 6. Impossible toy task to illustrate "cheating" with BN: *predicting whether the input is equal to the other sample in the batch.* During training, batch stats can be used to solve the task but the model fails at test time using global stats.

(a) *Cheating examples*   (b) *Cheating impairs BN models*

Fig. 7. *Anticipation (Cholec80):* (a) During training, BN models "cheat" by detecting instruments in later frames through batch stats. We visualize this by using batch stats at test time. "Cheating" still happens in batches with multiple or short sequences. (b) By using batch stats at test time, BN models achieve overly good scores *(cheating eval.)* but fail in a fair evaluation. Sampling multiple sequences per batch reduces but does not eliminate the effect.

The effect was most pronounced with sequences similar to the anticipation horizon (5 min), so we opt for partial freezing with training sequences of 256 frames (4 min 16 s).

Fig. 7(a) shows how anticipation predictions immediately improve in batches which contain an instrument occurrence at some point later in the batch. These examples visualize the "cheating" effect explicitly, not only indirectly through impact on performance. Likely, the model has learned to recognize the instrument and the change in feature activations becomes visible to previous frames through batch statistics. Namely, a sudden decrease of activations related to an instrument indicates that a higher mean was subtracted during normalization, so the instrument's occurrence later in the batch can be inferred. Interestingly, this effect is still visible (but less pronounced) in models trained on batches that contain multiple (4 × 256) or shorter (1 × 64) sequences.

In Fig. 7(b), we investigate how the "cheating" phenomenon affects anticipation performance. BN models are evaluated either using batch statistics (*cheating eval.*) or using global statistics, which would be the correct approach as future information is generally not available in an online anticipation setting. By using batch statistics at test time, the "cheating" model strongly (but unfairly) outperforms even BN-free end-to-end models. However, we can see that a stronger "cheating" effect (i.e. a lower error with *cheating eval.*, dotted line) leads to a higher error during correct evaluation (solid line). By sampling multiple sequences per batch (4 × 256), this effect is reduced and the BN model matches the performance of its GN counterpart. However, the gap between the two curves suggests that the model is still learning subpar features due to "cheating".

Hypothesis 10: End-to-end models outperform state of the art in anticipation. Finally, we compare our models and strategies to previous work in instrument anticipation (Rivoir et al., 2020; Yuan et al., 2021). In Table 6, we observe that ConvNeXt and FrozenBN outperform previous work. The GN model only performs well when
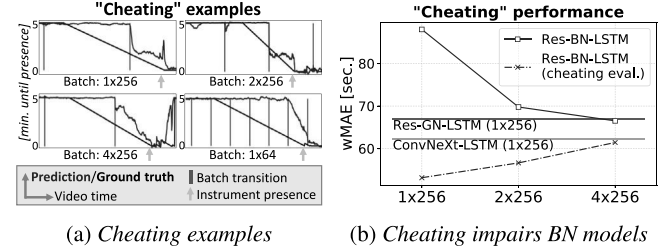
trained completely end-to-end but gives poor results with partial CNN freezing. As expected, BN models completely fail with single-sequence batches (1 × 64, 1 × 256) but also provide subpar performance with multiple long sequences (2 × 256, 4 × 256, 4 × 64). Likely, this is in part caused by BN's "cheating" (see Fig. 7(a)). However, other factors may have contributed since the partially frozen GN model underperforms as well. The end-to-end BN model trained on shorter 16-frame sequences performs surprisingly well. Likely, 16-second batches are too short for effective "cheating".

Across backbones, complete end-to-end training is more effective than freezing layers, possibly because anticipation is a more challenging task and requires more finetuning of CNN weights. Also, the CHE strategy outperforms CHT in this task. Presumably, the correlation between subsequent batches (and thus SGD steps) is not ideal for this more difficult task. Similarly, FrozenBN models underperformed in the CHT setting in phase recognition (Figs. 3(b), 5). Nevertheless, we provide strong evidence for the effectiveness of end-to-end learning also for anticipation and show that BN models are limited to short sequences and require careful engineering due to the "cheating" phenomenon.

5. Limitations and scope

Our proposed strategies aim at increasing temporal context, and thus seem mostly applicable to tasks where this is relevant, e.g. long videos and workflow-based tasks such as cooking or other complex activities. Its impact on other video tasks remains open. Further, our

Table 6
Comparison of anticipation methods (Cholec80). Scores (mean+std. over runs) which outperform state of the art are highlighted (**bold**).

Anticipation error ↓ (wMAE in s)					
SOTA	End-to-end AlexNet-LSTM (Rivoir et al., 2020)				66.02(±0.85)
	2-stage ResNet-TCN (Yuan et al., 2021) (variant with only visual features)				65.34(±0.59)
CNN–LSTM:		Res-BN	Res-FrozenBN	Res-GN	ConvNeXt
2-stage		65.54(±0.21)	–	66.54(±0.31)	65.30(±0.43)
E2E	1 × 64, CHT	–	66.36(±1.30)	67.18(±1.39)	**64.74(±0.12)**
	1 × 64, CHE	83.78(±3.29)	**62.10(±0.06)**	64.36(±0.28)	**61.72(±0.28)**
	4 × 16, CHE	**62.06(±0.95)**	–	–	–
Partial freezing	1 × 256, CHE	88.02(±16.58)	**63.10(±0.31)**	67.00(±0.87)	**62.34(±0.33)**
	2 × 256, CHE	69.78(±0.69)	–	–	–
	4 × 256, CHE	66.52(±0.51)	–	–	–
	4 × 64, CHE	65.30(±0.50)	–	–	–

experiments focus on 2D backbones, due to limitations in the surgical domain, and LSTMs, to show the effectiveness of simple, state-based end-to-end models for online tasks. While the incompatibility of BN and single-sequence batches is general to the video setting, investigating the feasibility and effectiveness of end-to-end approaches for other domains, backbones and temporal models requires further work and falls outside the scope of our study. Liu et al. (2022a) recently provided an extensive, insightful study on end-to-end temporal action detection with a wider range of models. However, BN issues are not discussed, although the common FrozenBN solution is utilized. Our work, on the other hand, aims at providing a deeper understanding of BN's limitations in long-video settings but restricts practical implications to a smaller range of models.

Nevertheless, we provide some hypotheses and ideas for future directions regarding other architectures: *(a)* 3D-CNNs do not differ in their use of normalization layers and thus BN issues and its solutions would likely be similar. Yet, available, non-causal 3D-CNNs pose practical limitations for online tasks since popular sliding-window approaches would pose strong memory constraints. *(b)* 2D Vision-Transformer backbones could be used in place of BN-free CNNs since they typically do not contain BatchNorm. Thus, we believe the increasing popularity of Transformers in computer vision represents an opportunity for end-to-end video learning. *(c)* For temporal modeling, CHT seemed to be an important component for pushing performance past SOTA models in our experiments and this strategy is only directly applicable in RNNs. However, for other temporal models like TCNs or Transformers, similar strategies could be conceivable by carrying features of convolutional or attention layers across batches. This could be an interesting direction for future work.

6. Conclusion

We investigate the limitations of BatchNorm for end-to-end video learning, which are especially relevant in surgical workflow tasks where CNN backbones require finetuning. In a detailed literature review, we reveal how research has circumvented these problems by moving towards complex, multi-stage approaches. We identify situations where BN models fail and how it affects end-to-end training strategies in simple CNN–LSTMs. Specifically, BN restricts the length of training sequences. In online settings, can interfere with hidden-state handling of LSTMs as well as leak information from future frames. We show that the latter can be used by models to "cheat" in anticipation tasks. We also find that the common ad-hoc solution FrozenBN is only effective in restricted settings. Instead, even simple CNN–LSTMs achieve state-of-the-art performance in three surgical workflow tasks by avoiding BatchNorm entirely and thus enabling longer temporal contexts.

We believe a comprehensive and deeper understanding of BN's limitations in video tasks is crucial for future research in surgical workflow analysis. End-to-end learning with simple models appears favorable over complex, multi-stage approaches but we hypothesize

that BN issues have silently hidden its advantages. Moving forward, the field could benefit from reconsidering end-to-end approaches and further investigating BN-free backbones. In natural-video tasks like action segmentation, where end-to-end learning currently plays only a minor role, re-evaluating training strategies could potentially be valuable as well.

CRediT authorship contribution statement

Dominik Rivoir: Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Isabel Funke:** Conceptualization, Writing – review & editing. **Stefanie Speidel:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dominik Rivoir reports financial support was provided by German Research Foundation (DFG, Deutsche Forschungsgemeinschaft). Stefanie Speidel reports a relationship with The Medical Image Computing and Computer Assisted Intervention Society that includes: board membership. Stefanie Speidel has been a Co-Guest Editor Special Issue for MICCAI 2022 and 2023.

Data availability

A link to the code associated with this work is provided in the article.

Appendix A. Results for surgical phase recognition

Tables A.7 and A.8 provide an overview of accuracy and F1 scores for surgical phase recognition. The accuracy scores were presented in the form of figures in the main paper (Figs. 2 & 3).

In Table A.9, we provide additional evidence that models with FrozenBN exhibit more instability during training.

Table A.7

Accuracy scores (mean + sample std. over runs) for surgical phase recognition on TeCNO's (Czempiel et al., 2020) split (40 train. videos, 8 val., 32 test) used for our main analysis. Per run, accuracy is computed per video and then averaged. We **highlight** scores which outperform all 2-stage approaches by at least its sample standard deviation. Note that these results were presented in the form of figures in the main paper.

Accuracy (video-wise) ↑	BN	FrozenBN	GN	ConvNeXt	Fig.
2-Stage (CNN-...)					
LSTM	89.1(±0.5)	–	87.2(±0.7)	87.8(±0.3)	
TCN (*TeCNO reimpl.*)	*89.4(±0.8)	–	88.4(±0.8)	88.2(±0.1)	2
TeCNO (Czempiel et al., 2020) (*original*)	88.56(±0.27)	–	–	–	
End-to-end (CNN–LSTM)					
8 × 8, sliding eval.	85.9(±0.1)	85.2(±0.6)	83.0(±2.2)	85.9(±0.2)	
4 × 16, sliding eval.	86.5(±1.4)	86.1(±1.2)	85.8(±0.7)	88.0(±0.9)	3(a)
2 × 32, sliding eval.	86.5(±0.4)	86.5(±1.2)	87.5(±2.3)	87.9(±0.5)	
1 × 64, sliding eval.	57.2(±4.2)	88.4(±1.9)	88.4(±0.8)	89.3(±1.2)	
8 × 8, carry-hidden eval.	89.4(±0.4)	89.0(±0.4)	86.8(±1.3)	88.1(±0.2)	
4 × 16, carry-hidden eval.	89.6(±1.0)	89.4(±1.0)	87.9(±0.8)	89.1(±1.1)	3(b)
2 × 32, carry-hidden eval.	88.2(±0.7)	89.4(±1.0)	88.9(±2.3)	88.6(±0.6)	
1 × 64, carry-hidden eval.	59.7(±3.8)	89.6(±1.1)	89.3(±0.4)	89.5(±1.1)	
1 × 64, carry-hidden train.	55.6(±3.1)	89.4(±0.4)	**90.5(±0.9)**	**92.3(±1.5)**	2 & 3(b)
Partially frozen (CNN–LSTM)					
4 × 64, carry-hidden eval.	90.1(±2.2)	–	–	–	
2 × 128, carry-hidden eval.	88.5(±1.6)	–	–	–	3(c)
1 × 256, carry-hidden eval.	76.7(±1.5)	90.0(±0.7)	**91.6(±0.5)**	90.6(±2.6)	
1 × 64, carry-hidden train.	–	**92.4(±0.4)**	89.7(±1.8)	**93.3(±0.3)**	
1 × 128, carry-hidden train.	–	**92.0(±0.5)**	**92.3(±0.5)**	**92.5(±1.2)**	3(c)
1 × 256, carry-hidden train.	–	**92.1(±0.4)**	**92.6(±0.5)**	**92.6(±0.1)**	

Table A.8

F1 scores (mean + sample std. over runs) for surgical phase recognition on TeCNO's (Czempiel et al., 2020) split (40 train. videos, 8 val., 32 test) used for our main analysis. Per training run, we compute the frame-based confusion matrix over the entire test set, compute a single F1 score per phase and average over phases. This strategy deviates from video-wise approaches in previous work but is less ambiguous and gives smoother scores in edge cases. We **highlight** scores which outperform all 2-stage approaches by at least its sample standard deviation.

F1 (frame-wise) ↑	BN	FrozenBN	GN	ConvNeXt	Fig.
2-Stage					
LSTM	83.2(±0.9)	–	80.2(±1.2)	81.0(±0.2)	2
TCN	83.4(±1.2)	–	81.0(±1.0)	81.7(±0.6)	
End-to-end (CNN–LSTM)					
8 × 8, sliding eval.	79.3(±0.9)	78.8(±0.2)	74.6(±2.1)	78.4(±1.2)	
4 × 16, sliding eval.	80.4(±1.7)	80.4(±0.8)	78.8(±0.7)	82.3(±1.0)	3(a)
2 × 32, sliding eval.	79.1(±0.9)	81.1(±0.5)	81.5(±2.7)	81.9(±1.3)	
1 × 64, sliding eval.	36.3(±3.1)	83.2(±1.4)	81.8(±0.7)	84.0(±1.0)	
8 × 8, carry-hidden eval.	83.2(±1.0)	82.8(±0.8)	79.4(±0.8)	81.9(±0.6)	
4 × 16, carry-hidden eval.	83.6(±1.6)	83.2(±0.9)	81.5(±0.5)	83.7(±1.2)	3(b)
2 × 32, carry-hidden eval.	81.3(±0.9)	82.8(±1.1)	82.9(±2.9)	83.2(±0.8)	
1 × 64, carry-hidden eval.	37.1(±3.0)	82.8(±1.9)	82.9(±0.5)	**84.5(±0.9)**	
1 × 64, carry-hidden train.	40.0(±4.1)	82.2(±0.2)	83.9(±0.7)	**86.2(±1.7)**	2 & 3(b)
Partially frozen (CNN–LSTM)					
4 × 64, carry-hidden eval.	84.5(±1.4)	–	–	–	
2 × 128, carry-hidden eval.	79.8(±2.6)	–	–	–	3(c)
1 × 256, carry-hidden eval.	58.8(±0.9)	82.9(±2.2)	83.7(±0.9)	**85.8(±2.4)**	
1 × 64, carry-hidden train.	–	**85.3(±1.3)**	82.4(±1.0)	**87.4(±0.7)**	
1 × 128, carry-hidden train.	–	**85.0(±0.5)**	83.8(±1.7)	**87.5(±0.9)**	3(c)
1 × 256, carry-hidden train.	–	**85.5(±0.6)**	**84.8(±0.4)**	**87.1(±0.5)**	

Appendix B. Results for online action segmentation

Tables B.10 and B.11 show our results on the GTEA (Fathi et al., 2011) and 50Salads (Stein and McKenna, 2013) datasets. These are the same results which were presented as bar plots in Fig. 5 of the paper. Differently from most previous work, we formulate action segmentation as an online task on these datasets. We use the suggested splits for K-fold cross validation but for each training run, we use one set for validation (model selection), one set for testing and the remaining for training. We use the same hyperparameters as for surgical phase recognition (see Appendix E).

Appendix C. Model architectures

Fig. C.8 shows the network architectures used in our experiments. For the GroupNorm-based Resnet-50, we use GroupNorm with 32 groups since this performed best in the original paper (Wu and He, 2018) and has a pretrained model available.[3] For 2-stage TCN models, we follow TeCNO (Czempiel et al., 2020) as close as possible. Although

---

[3] https://github.com/ppwwyyxx/GroupNorm-reproduce/releases/tag/v0.1.

Table A.9

FrozenBN models exhibit more instability during training than GN models. Initial learning rates of FrozenBN models often had to be reduced in order to prevent model collapse (red), but higher learning rates were preferable in cases without collapse.

Learning rate	Accuracy (video-wise) ↑			
	FrozenBN		GN	
	E2E (1 × 64)			
	CHE	CHT	CHE	CHT
10^{-6}	88.2(±0.1)	**89.4(±0.4)**	83.7(±2.3)	88.4(±0.2)
10^{-5}	**89.6(±1.1)**	39.8(±0.0)	**89.3(±0.4)**	**90.5(±0.9)**
	Partial freezing (1 × 256)			
	CHE	CHT	CHE	CHT
10^{-5}	**90.0(±0.7)**	**92.1(±0.4)**	86.9(±0.5)	89.5(±0.7)
10^{-4}	39.8(±0.0)	39.8(±0.0)	**91.1(±0.9)**	**92.6(±0.5)**

Table B.10

Accuracy scores for *online* action segmentation on the *GTEA* dataset (Fathi et al., 2011). We use the suggested splits for 4-fold cross validation. Note that most works propose offline methods. For each backbone, we **highlight** the training strategy (CHE vs. CHT) which performs better. Note that these results were presented in the form of a bar plot in the main paper.

GTEA				
Accuracy (video-wise) ↑				
	BN	FrozenBN	GN	ConvNeXt
2-Stage				
LSTM	60.1	–	–	–
TCN	63.2	–	–	–
End-to-end, 1 × 64				
CHE	–	**66.6**	63.9	63.5
CHT	43.6	64.3	**66.8**	**68.9**

Table B.11

Accuracy scores for *online* action segmentation on the *50Salads* dataset (Stein and McKenna, 2013). We use the suggested splits for 5-fold cross validation. Note that most works propose offline methods. For each backbone, we **highlight** the training strategy (CHE vs. CHT) which performs better. Note that these results were presented in the form of a bar plot in the main paper.

50Salads				
Accuracy (video-wise) ↑				
	BN	FrozenBN	GN	ConvNeXt
2-Stage				
LSTM	69.4	–	–	–
TCN	72.0	–	–	–
End-to-end, 1 × 64				
CHE	–	**77.9**	80.4	79.3
CHT	23.5	77.5	**83.8**	**83.3**

not reported in the paper, we follow their exact model configuration with 2 stages, 9 layers per stage and 64 feature channels per layer.[4]

Fig. C.9 shows which backbone blocks are frozen in the respective experiments. Note that with a BN-based backbone, batch statistics were still used for normalization during training. Using FrozenBN (i.e. using ImageNet's global stats) in frozen layers yielded similar results.

Appendix D. Task formulations

D.1. Surgical phase recognition

Surgical phase recognition on the Cholec80 dataset (Twinanda et al., 2016) is a dense temporal segmentation task with 7 phases and is

---

therefore a simple multi-class classification when viewed at the frame level. Given a frame x with corresponding class label $c \in \{1, \ldots, 7\}$ and a softmax-activated neural network f which outputs a 7D vector per input image, we use the standard cross-entropy loss for optimization:

$$L_{phase}(x, c) = -\log f(x)[c] \qquad (D.1)$$

D.2. Anticipation of surgical instrument usage

For instrument anticipation we follow the task formulation from previous work (Rivoir et al., 2020). We anticipate 5 different instruments on the Cholec80 dataset, where the main task is to predict the remaining duration until occurrence of each instrument at each time point up to a horizon of 5 min. As an auxiliary classification task, we also predict whether each instrument is present and whether or not it will appear within the next 5 min (i.e. inside or outside of horizon).

Given a frame x and the corresponding remaining durations t_i for each instrument $i \in \{1, \ldots, 5\}$ and the horizon $h = 5$, the ground-truth value for the main regression task is defined as

$$y_i = \min(t_i, h) . \qquad (D.2)$$

The corresponding ground-truth class for the auxiliary classification task is

$$c_i = \begin{cases} 1, & y_i = 0 \text{ (instrument present)} \\ 2, & 0 < y_i < h \text{ (inside horizon)} \\ 3, & y_i = h \text{ (outside horizon)} \end{cases} . \qquad (D.3)$$

We then jointly optimize both tasks with the SmoothL1 (Yengera et al., 2018) loss for regression and cross entropy for classification:

$$L_{ant}(x, y) = \sum_{i=1}^{5} \text{SmoothL1}(f_{reg}^i(x), y_i) - \lambda \log f_{cls}^i(x)[c_i] , \qquad (D.4)$$

where f_{reg}^i and f_{cls}^i are the regression and classification predictions of a neural network for instrument i and $\lambda = 0.01$ is a scaling factor for the auxiliary task.

Appendix E. Training details & hyperparameters

Tables E.12–E.14 summarize our training hyperparameters. Some of our choices might require clarification:

1. In 'partially frozen' models for phase recognition, we trained for 100 epochs on the 40/8/32 split, but 200 epochs on the 32/8/40 split for state-of-the-art comparison due to the smaller training set. This was only done for efficiency reasons, since 100 epochs seemed to be enough with 40 train videos. The 'partial freezing' models did not deteriorate with longer training duration.

2. When training only the backbone (for 2-stage approaches), we shortened the definition of an epoch, since the CNN overfit too quickly on the image task and would provide bad features for the temporal model in the subsequent stage.

3. Despite our use of different learning rates, our decision rule was very simple: We use a default 10^{-5} for models where all parameters are trained (*End-to-end* & *Backbone only*) and increased the learning by a factor of 10 for shallower models (*Partially frozen*). For training only the temporal models in 2-stage approaches (*Temporal only*), we follow TeCNO (Czempiel et al., 2020) and use $5 \cdot 10^{-4}$, which seemed to perform best in our experiments, too.

4. Except for the number of epochs, we use exactly the same hyperparameters for phase recognition and anticipation. Convergence was much slower in anticipation.
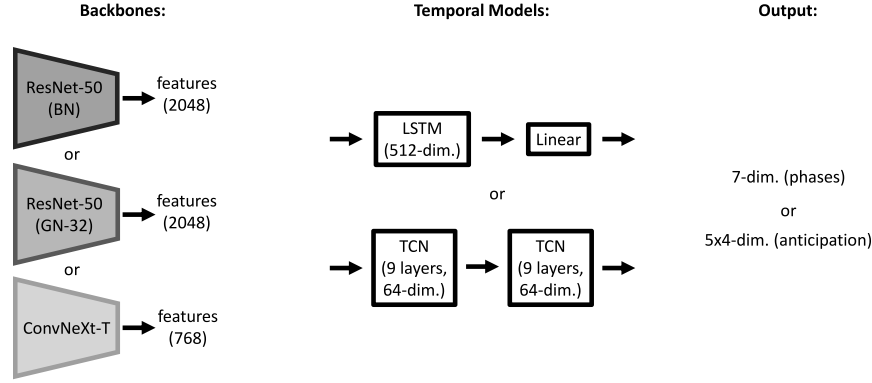
Backbones: Temporal Models: Output:

Fig. C.8. Overview of model architectures used in our experiments. For instrument anticipation, we follow the original task formulation (Rivoir et al., 2020). For each of 5 instruments, 4 prediction values are required (1 for duration prediction and 3 for the auxiliary classification task).
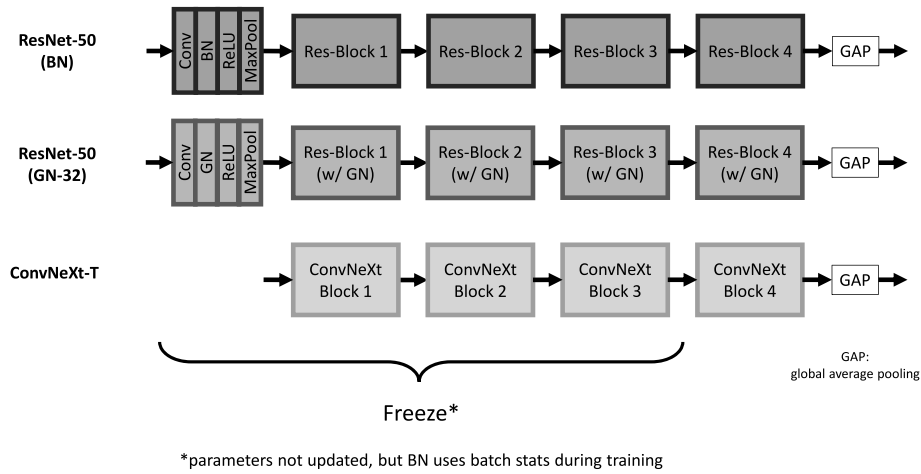
Fig. C.9. How backbones were frozen in the 'partially frozen' models. Note that with a BN-based backbone, batch statistics were still used for normalization during training. Using FrozenBN (i.e. using ImageNet's global stats) in frozen layers yielded similar results.

Table E.12

Overview of global hyperparameters (used in all settings).

Optimizer	AdamW
Weight decay	0.01
Image size	216 × 384
Data aug.	Shift, Scale, Crop, RGB shift, Brightness shift, Contrast shift

Table E.13

Overview of hyperparameters which depend on the training strategy or task (for Cholec80).

Training type	Epochs		Learn. rate	Batch size	Batches per epoch
	Phase	Antic.	Both tasks		
Backbone only	50	300	10^{-5}	32	#frames/256
Temporal only	100	100	$5 \cdot 10^{-4}$	Whole video	#videos
End-to-end	200	300	$10^{-5\,b}$	64	#frames/batch size
Partially frozen	200^a	300	$10^{-4\,b}$	64,128,256	#frames/batch size

[a] 100 on the 40/8/32 split. Solely for efficiency reasons. Longer training did not decrease performance.

[b] FrozenBN models which collapsed under these parameters used a reduced learning rate by factor 10 (E2E/CHT, frozen/CHE, frozen/CHT, on Cholec80 and 50Salads).

E.1. Varying hyperparameters on other datasets

When retraining models on datasets other than Cholec80, we had to make changes to some hyperparameters:

50Salads: No changes were made.

GTEA: We did not have to reduce learning rates for FrozenBN models. All other hyperparameters are identical to Cholec80.

AutoLaparo: We increased the learning rate of partially frozen ConvNeXt models by a factor of 5 (from 10^{-4} to $5 \cdot 10^{-4}$). In all other settings, this was not effective. Further, we found that accuracy was not a suitable metric for model selection on AutoLaparo due to the highly imbalanced dataset. Sometimes, models would achieve very high validation accuracy but low phase-based scores early during training and thus would lead to poor models being selected for testing. Instead, we used frame-based F1 for model selection.

CATARACTS: We increased learning rates of all ConvNeXt and BN models by a factor of 5 (E2E: $5 \cdot 10^{-5}$, partial freezing: $5 \cdot 10^{-4}$). For GN, this was not effective, so we kept the initial learning rates.

Appendix F. Metrics

Mean video-wise accuracy: We compute the mean video-wise accuracy, i.e. we compute one accuracy score per video and average over videos.

Mean video-wise balanced accuracy: We compute one balanced accuracy score per video and average over videos. Undefined cases do not occur here.

Precision, Recall, Jaccard, F1: Per class, we compute true positives, false positives and false negatives over the entire test set. Then a single score is computer per class and averaged over classes. Only precision can encounter undefined values in this case, which we set to zero.

Table E.14
Datasets.

	FPS	Image resolution	Classes	Videos
Cholec80 (Twinanda et al., 2016)	1	216 × 384	7	80
AutoLaparo (Wang et al., 2022)	1	216 × 384	7	21
CATARACTS (Zisimopoulos et al., 2018)	5	216 × 384	19	50
GTEA (Fathi et al., 2011)	15	216 × 288	11	28
50Salads (Stein and McKenna, 2013)	5	216 × 288	19	50

Relaxed metrics for SOTA comparison: Some methods (including the current state-of-the-art methods Trans-SVNet Gao et al., 2021 and TMRNet Jin et al., 2021) report relaxed versions of metrics on the CHolec80 dataset, which were proposed in the *M2CAI 2016 Challenge* (Stauder et al., 2016). Here, errors around phase transitions are ignored due to their ambiguity. Implementation details can be found in TMRNet's code.[5]

Weighted Mean Absolute Error (wMAE): We follow the original instrument anticipation approach (Rivoir et al., 2020) and report *wMAE* as our main metric. This is the mean of the frame-wise mean absolute errors inside *(inMAE)* and outside *(outMAE)* of the horizon. Specifically for each instrument, predictions on all test videos are concatenated into a single sequence, to compute single *inMAE* and *outMAE* (and thus *wMAE*) scores per instrument, which are then averaged.

References

Abu Farha, Y., Ke, Q., Schiele, B., Gall, J., 2020. Long-term anticipation of activities with cycle consistency. In: DAGM German Conference on Pattern Recognition. Springer, pp. 159–173.

Aksamentov, I., Twinanda, A.P., Mutter, D., Marescaux, J., Padoy, N., 2017. Deep neural networks predict remaining surgery duration from cholecystectomy videos. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 586–593.

Andreux, M., Terrail, J.O.d., Beguier, C., Tramel, E.W., 2020. Siloed federated learning for multi-centric histopathology datasets. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. Springer, pp. 129–139.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv: 1607.06450.

Bano, S., Vasconcelos, F., Vander Poorten, E., Vercauteren, T., Ourselin, S., Deprest, J., Stoyanov, D., 2020. FetNet: a recurrent convolutional network for occlusion identification in fetoscopic videos. Int. J. Comput. Assist. Radiol. Surg. 15 (5), 791–801.

Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q., 2018. Understanding batch normalization. Adv. Neural Inf. Process. Syst. 31.

Bodenstedt, S., Rivoir, D., Jenke, A., Wagner, M., Breucha, M., Müller-Stich, B., Mees, S.T., Weitz, J., Speidel, S., 2019a. Active learning using deep Bayesian networks for surgical workflow analysis. Int. J. Comput. Assist. Radiol. Surg. 14 (6), 1079–1087.

Bodenstedt, S., Wagner, M., Mündermann, L., Kenngott, H., Müller-Stich, B., Breucha, M., Mees, S.T., Weitz, J., Speidel, S., 2019b. Prediction of laparoscopic procedure duration using unlabeled, multimodal sensor data. Int. J. Comput. Assist. Radiol. Surg. 14 (6), 1089–1095.

Brock, A., De, S., Smith, S.L., Simonyan, K., 2021. High-performance large-scale image recognition without normalization. In: International Conference on Machine Learning. PMLR, pp. 1059–1071.

Cai, D., Yao, A., Chen, Y., 2021. Dynamic normalization and relay for video action recognition. Adv. Neural Inf. Process. Syst. 34, 11026–11040.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308.

Chen, H.-B., Li, Z., Fu, P., Ni, Z.-L., Bian, G.-B., 2022. Spatio-temporal causal transformer for multi-grained surgical phase recognition. In: 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society. EMBC, IEEE, pp. 1663–1666.

Czempiel, T., Paschali, M., Keicher, M., Simson, W., Feussner, H., Kim, S.T., Navab, N., 2020. Tecno: Surgical phase recognition with multi-stage temporal convolutional networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 343–352.

Czempiel, T., Paschali, M., Ostler, D., Kim, S.T., Busam, B., Navab, N., 2021. Opera: Attention-regularized transformers for surgical phase recognition. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 604–614.

Czempiel, T., Sharghi, A., Paschali, M., Navab, N., Mohareri, O., 2022. Surgical workflow recognition: From analysis of challenges to architectural study. In: European Conference on Computer Vision. Springer, pp. 556–568.

De, S., Smith, S., 2020. Batch normalization biases residual blocks towards the identity function in deep networks. Adv. Neural Inf. Process. Syst. 33, 19964–19975.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Eun, H., Moon, J., Park, J., Jung, C., Kim, C., 2020. Learning to discriminate information for online action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 809–818.

Farha, Y.A., Gall, J., 2019. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3575–3584.

Fathi, A., Ren, X., Rehg, J.M., 2011. Learning to recognize objects in egocentric activities. In: CVPR 2011. IEEE, pp. 3281–3288.

Funke, I., Bodenstedt, S., Oehme, F., von Bechtolsheim, F., Weitz, J., Speidel, S., 2019. Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 467–475.

Furnari, A., Farinella, G.M., 2020. Rolling-unrolling lstms for action anticipation from first-person video. IEEE Trans. Pattern Anal. Mach. Intell. 43 (11), 4021–4036.

Gao, X., Jin, Y., Long, Y., Dou, Q., Heng, P.-A., 2021. Trans-svnet: accurate phase recognition from surgical videos via hybrid embedding aggregation transformer. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 593–603.

Gao, J., Yang, Z., Nevatia, R., 2017. Red: Reinforced encoder-decoder networks for action anticipation. arXiv preprint arXiv:1707.04818.

Ghoddoosian, R., Dwivedi, I., Agarwal, N., Choi, C., Dariush, B., 2022. Weakly-supervised online action segmentation in multi-view instructional videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13780–13790.

Girdhar, R., Grauman, K., 2021. Anticipative video transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13505–13515.

He, Z., Mottaghi, A., Sharghi, A., Jamal, M.A., Mohareri, O., 2022. An empirical study on activity recognition in long surgical videos. In: Machine Learning for Health. PMLR, pp. 356–372.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Henaff, O., 2020. Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. PMLR, pp. 4182–4192.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hoffer, E., Hubara, I., Soudry, D., 2017. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. Adv. Neural Inf. Process. Syst. 30.

Huang, Y., Sugano, Y., Sato, Y., 2020. Improving action segmentation via graph-based temporal reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14024–14034.

Ioffe, S., 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. Adv. Neural Inf. Process. Syst. 30.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. PMLR, pp. 448–456.

Ishikawa, Y., Kasai, S., Aoki, Y., Kataoka, H., 2021. Alleviating over-segmentation errors by detecting action boundaries. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2322–2331.

Jin, Y., Dou, Q., Chen, H., Yu, L., Heng, P.-A., 2016. EndoRCN: recurrent convolutional networks for recognition of surgical workflow in cholecystectomy procedure video. IEEE Trans. Med. Imaging.

Jin, Y., Dou, Q., Chen, H., Yu, L., Qin, J., Fu, C.-W., Heng, P.-A., 2017. SV-RCNet: workflow recognition from surgical videos using recurrent convolutional network. IEEE Trans. Med. Imaging 37 (5), 1114–1126.

---

[5] https://github.com/YuemingJin/TMRNet/blob/main/code/eval/result/matlab-eval/.

Jin, Y., Li, H., Dou, Q., Chen, H., Qin, J., Fu, C.-W., Heng, P.-A., 2020. Multi-task recurrent convolutional network with correlation loss for surgical video analysis. Med. Image Anal. 59, 101572.

Jin, Y., Long, Y., Chen, C., Zhao, Z., Dou, Q., Heng, P.-A., 2021. Temporal memory relation network for workflow recognition from surgical video. IEEE Trans. Med. Imaging 40 (7), 1911–1923.

Kadkhodamohammadi, A., Luengo, I., Stoyanov, D., 2022. PATG: position-aware temporal graph networks for surgical phase recognition on laparoscopic videos. Int. J. Comput. Assist. Radiol. Surg. 17 (5), 849–856.

Kannan, S., Yengera, G., Mutter, D., Marescaux, J., Padoy, N., 2019. Future-state predicting LSTM for early surgery type recognition. IEEE Trans. Med. Imaging 39 (3), 556–566.

Ke, Q., Fritz, M., Schiele, B., 2019. Time-conditioned action anticipation in one shot. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9925–9934.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems. Vol. 25, Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Labatie, A., Masters, D., Eaton-Rosen, Z., Luschi, C., 2021. Proxy-normalizing activations to match batch normalization while removing batch dependence. Adv. Neural Inf. Process. Syst. 34.

Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D., 2017. Temporal convolutional networks for action segmentation and detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR.

Liu, X., Bai, S., Bai, X., 2022a. An empirical study of end-to-end temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20010–20019.

Liu, Y., Boels, M., Garcia-Peraza-Herrera, L.C., Vercauteren, T., Dasgupta, P., Granados, A., Ourselin, S., 2023a. LoViT: Long video transformer for surgical phase recognition. arXiv preprint arXiv:2305.08989.

Liu, H., Brock, A., Simonyan, K., Le, Q., 2020. Evolving normalization-activation layers. Adv. Neural Inf. Process. Syst. 33, 13539–13550.

Liu, Y., Huo, J., Peng, J., Sparks, R., Dasgupta, P., Granados, A., Ourselin, S., 2023b. SKiT: a fast key information video transformer for online surgical phase recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21074–21084.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022b. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H., 2022c. Video swin transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3202–3211.

Maier-Hein, L., Eisenmann, M., Sarikaya, D., März, K., Collins, T., Malpani, A., Fallert, J., Feussner, H., Giannarou, S., Mascagni, P., et al., 2022. Surgical data science–from concepts toward clinical translation. Med. Image Anal. 76, 102306.

Maier-Hein, L., Vedula, S.S., Speidel, S., Navab, N., Kikinis, R., Park, A., Eisenmann, M., Feussner, H., Forestier, G., Giannarou, S., et al., 2017. Surgical data science for next-generation interventions. Nat. Biomed. Eng. 1 (9), 691–696.

Marafioti, A., Hayoz, M., Gallardo, M., Márquez Neila, P., Wolf, S., Zinkernagel, M., Sznitman, R., 2021. CataNet: Predicting remaining cataract surgery duration. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 426–435.

Nwoye, C.I., Mutter, D., Marescaux, J., Padoy, N., 2019. Weakly supervised convolutional LSTM approach for tool tracking in laparoscopic videos. Int. J. Comput. Assist. Radiol. Surg. 14 (6), 1059–1067.

Pham, Q., Liu, C., Hoi, S., 2022. Continual normalization: Rethinking batch normalization for online continual learning. arXiv preprint arXiv:2203.16102.

Rivoir, D., Bodenstedt, S., Bechtolsheim, F.v., Distler, M., Weitz, J., Speidel, S., 2019. Unsupervised temporal video segmentation as an auxiliary task for predicting the remaining surgery duration. In: OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging. Springer, pp. 29–37.

Rivoir, D., Bodenstedt, S., Funke, I., Bechtolsheim, F.v., Distler, M., Weitz, J., Speidel, S., 2020. Rethinking anticipation tasks: Uncertainty-aware anticipation of sparse surgical instrument usage for context-aware assistance. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 752–762.

Salimans, T., Kingma, D.P., 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Adv. Neural Inf. Process. Syst. 29.

Sener, F., Singhania, D., Yao, A., 2020. Temporal aggregate representations for long-range video understanding. In: European Conference on Computer Vision. Springer, pp. 154–171.

Shen, S., Yao, Z., Gholami, A., Mahoney, M., Keutzer, K., 2020. Powernorm: Rethinking batch normalization in transformers. In: International Conference on Machine Learning. PMLR, pp. 8741–8751.

Shi, X., Jin, Y., Dou, Q., Heng, P.-A., 2020. LRTD: long-range temporal dependency based active learning for surgical workflow recognition. Int. J. Comput. Assist. Radiol. Surg. 15 (9), 1573–1584.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singh, S., Krishnan, S., 2020. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11237–11246.

Singh, S., Shrivastava, A., 2019. Evalnorm: Estimating batch normalization statistics for evaluation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3633–3641.

Stauder, R., Ostler, D., Kranzfelder, M., Koller, S., Feußner, H., Navab, N., 2016. The TUM LapChole dataset for the M2CAI 2016 workflow challenge. arXiv preprint arXiv:1610.09278.

Stein, S., McKenna, S.J., 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 729–738.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114.

Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., De Mathelin, M., Padoy, N., 2016. Endonet: a deep architecture for recognition tasks on laparoscopic videos. IEEE Trans. Med. Imaging 36 (1), 86–97.

Twinanda, A.P., Yengera, G., Mutter, D., Marescaux, J., Padoy, N., 2018. RSDNet: Learning to predict remaining surgery duration from laparoscopic videos without manual annotations. IEEE Trans. Med. Imaging 38 (4), 1069–1078.

Ulyanov, D., Vedaldi, A., Lempitsky, V., 2016. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, Z., Gao, Z., Wang, L., Li, Z., Wu, G., 2020. Boundary-aware cascade networks for temporal action segmentation. In: European Conference on Computer Vision. Springer, pp. 34–51.

Wang, Z., Lu, B., Long, Y., Zhong, F., Cheung, T.-H., Dou, Q., Liu, Y., 2022. Autolaparo: A new dataset of integrated multi-tasks for image-guided surgical automation in laparoscopic hysterectomy. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 486–496.

Wu, Y., He, K., 2018. Group normalization. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 3–19.

Wu, Y., Johnson, J., 2021. Rethinking" batch" in batchnorm. arXiv preprint arXiv:2105.07576.

Xu, M., Gao, M., Chen, Y.-T., Davis, L.S., Crandall, D.J., 2019. Temporal recurrent networks for online action detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5532–5541.

Yan, J., Wan, R., Zhang, X., Zhang, W., Wei, Y., Sun, J., 2020. Towards stabilizing batch statistics in backward propagation of batch normalization. arXiv preprint arXiv:2001.06838.

Yang, X., Mirmehdi, M., Burghardt, T., 2021. Back to the future: Cycle encoding prediction for self-supervised video representation learning. In: The 32nd British Machine Vision Conference.

Yao, Z., Cao, Y., Lin, Y., Liu, Z., Zhang, Z., Hu, H., 2021a. Leveraging batch normalization for vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 413–422.

Yao, Z., Cao, Y., Zheng, S., Huang, G., Lin, S., 2021b. Cross-iteration batch normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12331–12340.

Yengera, G., Mutter, D., Marescaux, J., Padoy, N., 2018. Less is more: Surgical phase recognition with less annotations through self-supervised pre-training of CNN-LSTM networks. arXiv preprint arXiv:1805.08569.

Yi, F., Wen, H., Jiang, T., 2021. ASFormer: Transformer for action segmentation. In: The British Machine Vision Conference. BMVC.

Yi, F., Yang, Y., Jiang, T., 2022. Not end-to-end: Explore multi-stage architecture for online surgical phase recognition. In: Proceedings of the Asian Conference on Computer Vision. pp. 2613–2628.

Yuan, K., Holden, M., Gao, S., Lee, W.-S., 2021. Surgical workflow anticipation using instrument interaction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 615–625.

Zhang, Y., Bano, S., Page, A.-S., Deprest, J., Stoyanov, D., Vasconcelos, F., 2022. Large-scale surgical workflow segmentation for laparoscopic sacrocolpopexy. Int. J. Comput. Assist. Radiol. Surg. 1–11.

Zhang, B., Ghanem, A., Simes, A., Choi, H., Yoo, A., Min, A., 2021. Swnet: Surgical workflow recognition with deep convolutional network. In: Medical Imaging with Deep Learning. PMLR, pp. 855–869.

Zhao, Y., Krähenbühl, P., 2022. Real-time online video detection with temporal smoothing transformers. In: European Conference on Computer Vision. Springer, pp. 485–502.

Zhao, P., Xie, L., Zhang, Y., Wang, Y., Tian, Q., 2020. Privileged knowledge distillation for online action detection. arXiv preprint arXiv:2011.09158.

Zisimopoulos, O., Flouty, E., Luengo, I., Giataganas, P., Nehme, J., Chow, A., Stoyanov, D., 2018. Deepphase: surgical phase recognition in cataracts videos. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 265–272.