CHALLENGES OF DECOMPOSING TOOLS IN SURGICAL SCENES THROUGH DISENTANGLING THE LATENT REPRESENTATIONS

Sai Lokesh Gorantla, Raviteja Sista, Apoorva Srivastava, P P Chakrabarti, Debdoot Sheet
Indian Institute of Technology Kharagpur, India
g.sailokesh9@gmail.com, sista.raviteja@kgpian.iitkgp.ac.in,
apoorva.s.2311@gmail.com, {ppchak@cse, debdoot@ee}.iitkgp.ac.in

Utpal De
Nil Ratan Sircar Medical College and Hospital, India
utpalde9@gmail.com

ABSTRACT

Image generation through disentangling object representations is a critical area of research with significant potential. Disentanglement involves separating the representation of objects and their attributes, enabling greater control over the generated output. However, existing approaches are limited to disentangling only the objects' attributes and generating images with selected combinations of attributes. This study explores learning object-level disentanglement of semantically rich latent representation using von-Mises-Fisher (vMF) distributions. The proposed approach aims to disentangle compressed representations into object and background classes. The approach is tested on surgical scenes for disentanglement of tools and background information using the Cholec80 dataset. Achieving tool-background disentanglement provides an opportunity to generate rare and custom surgical scenes. However, the proposed method learns to disentangle representations based on pixel intensities. This study uncovers the challenges and shortfalls in achieving object-level disentanglement of the compressed representations using vMF distributions. The code for this study is available at https://github.com/it-is-lokesh/vMF-disentanglement-challenges.

1 INTRODUCTION

Convolutional neural networks (CNNs) have demonstrated remarkable success in performing tasks like classification, object detection, localization, etc. (Krichen, 2023). However, their ability to learn disentangled representations of objects in an image has vast potential and has been under-explored. A representation is a condensed, encoded, and structured summary of object-specific attributes. Disentanglement isolates object-specific representations of each object in an image into independent channels of the representational space, thus enabling a network to encapsulate semantically rich and interpretable object features. This ability has valuable applications, such as facilitating the generation of new images by selecting specific channels of the representational space. This enables the creation of synthetic datasets and advanced analytical applications such as focused or highlighted object tracking in real-time video feeds.

This work aims to learn disentangled representations of surgical scenes, as depicted in Fig. 1. It illustrates the hypothesis and overview of the proposed approach, which is designed to disentangle object/class representations into independent channels. For this work, two classes are chosen: tools (all surgical tools in the dataset), and background/non-tools (everything other than tools). The representations of non-tools and tools is denoted as $\mathbf{z}^{I}_{vMF}$ and $\mathbf{z}^{II}_{vMF}$ respectively (Fig. 1).

Recent advancements in disentangled representation learning have laid a promising foundation for this task. Feature disentanglement methods that use supervision through segmentation ground truth have successfully encapsulated object-specific information into separate latent channels (Tomar &

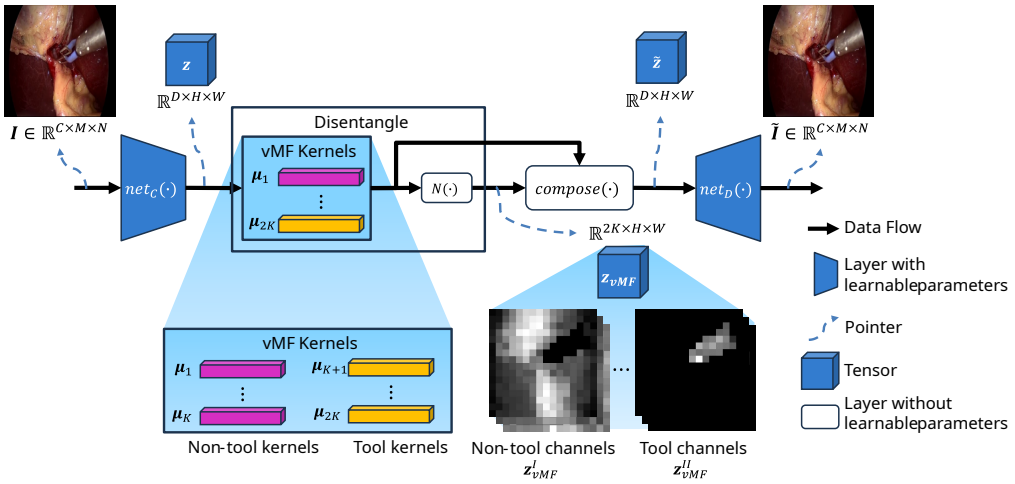


Figure 1: An overview of the proposed method for learning disentangled representations of surgical scenes. Details about the network modules are detailed in Sec. 2.

Rajagopalan, 2022; Liu et al., 2022). Generative adversarial networks (GAN) have shown decent progress in disentanging object-specific attributes (Zhu et al., 2020). In addition, Dombrowski et al. (2023) developed a pipeline to extract foreground and background masks utilizing a student-teacher based approach. Although these approaches have significantly progressed in learning disentangled representations, they rely heavily on segmentation ground truth. The current work focuses on achieving the same task without external supervision. Since obtaining high-quality annotations for medical data increases the cost of data procurement and compressed representations. Here, a convolution network as described in (Kakaiya et al., 2023; Raj et al., 2023) that produces semantically rich compressed representations is employed. To disentangle the compressed representations, learnable von-Mises-Fisher (vMF) distribution parameters are employed, similar to Liu et al. (2022).

2 Method

2.1 Network architecture

The proposed network consists of four modules; encoder block ($net_E(\cdot)$), disentanglement block ($disentangle(\cdot)$), composition block ($compose(\cdot)$), and decoder block ($net_D(\cdot)$). The encoder takes an image $\boldsymbol{I}$ as input and compresses it to obtain a latent representation $\mathbf{z}$. The obtained latent representation $\mathbf{z}$ is convolved with $2K$ vMF kernels, represented by $\boldsymbol{\mu}$, followed by channel-wise normalization($N(\cdot)$) to obtain $\mathbf{z}_{vMF}$. An affine transformation, termed as compose operation, is performed using $\boldsymbol{\mu} = \{\boldsymbol{\mu}^I, \boldsymbol{\mu}^{II}\}$ and $\mathbf{z}_{vMF}$ to obtain $\tilde{\mathbf{z}}$. The $\tilde{\mathbf{z}}$ is provided to the decoder as input to reconstruct the original image $\tilde{\boldsymbol{I}}$. The schematic representation of the network is shown in Fig. 1. Training the network consists of two stages, where the objective in Stage 1 is to learn representations of the background using K kernels ($\boldsymbol{\mu}^I$). In Stage 2, the objective is to learn representations of tools using another set of K kernels ($\boldsymbol{\mu}^{II}$).

2.2 Learning objective

The trainable parameters corresponding to the blocks $net_E(\cdot)$, the $net_D(\cdot)$ and the $disentangle(\cdot)$ are represented as $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$ and $\boldsymbol{\mu} = \{\boldsymbol{\mu}^I, \boldsymbol{\mu}^{II}\}$ respectively. The objective function employed for this task consists of three components: reconstruction loss $L_{rec}(\boldsymbol{I}, \tilde{\boldsymbol{I}})$, vMF loss $L_{vMF}(\boldsymbol{\mu}, \mathbf{z})$ and dissimilarity loss $L_{dis,s}(\boldsymbol{\mu})$ where s indicates the training stage. Mean absolute error (MAE) is used for reconstruction loss. The vMF loss is mathematically represented as $L_{vMF}(\boldsymbol{\mu}, \mathbf{z}) = -(HW)^{-1}\sum_i \max_j \boldsymbol{\mu}_j^I \mathbf{z}_i$ where i indicates the index of the feature vectors, j indicates the index of the vMF kernels, and H, W are the height and width of the latent representation.

The dissimilarity loss for Stage 1 is formulated as $L_{dis,1}(\boldsymbol{\mu}^I) = -\sum_{i=0}^{K-1}\sum_{j=i+1}^{K-1} \boldsymbol{\mu}_i^I \boldsymbol{\mu}_j^I$ and for Stage 2 the dissimilarity loss is given by $L_{dis,2}(\boldsymbol{\mu}) = -\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} \boldsymbol{\mu}_i^I \boldsymbol{\mu}_j^{II}$. The vMF loss and

dissimilarity loss are weighted by the parameters $\boldsymbol{\alpha}_s$ and $\boldsymbol{\beta}_s$ where s is the training stage. The vMF loss forces the kernels to be the cluster centers of the feature vectors (Kortylewski et al., 2020), thus capturing the intrinsic patterns within them. The dissimilarity loss is employed to maintain a degree of dissimilarity between the vMF kernels, preventing them from collapsing in the same direction. The learning objective for stage s is mathematically given as:

$$\boldsymbol{\mu}^*, \boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^* = \underset{\boldsymbol{\mu}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2}{\arg\min}\, L_{rec}(\boldsymbol{I}, \tilde{\boldsymbol{I}}) + \boldsymbol{\alpha}_s L_{vMF}(\boldsymbol{\mu}, \mathbf{z}) + \boldsymbol{\beta}_s L_{dis,s}(\boldsymbol{\mu}) \tag{1}$$

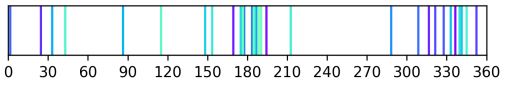
3 Experiment Results and Discussions

3.1 Experimental setup

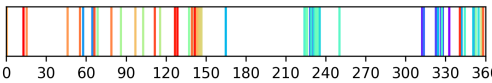
Cholec80 dataset (Twinanda et al., 2016), containing surgical scenes from 80 laparoscopic cholecystectomy videos, is used to test the proposed approach. Stage 1 training is performed with images where tools are absent, and only the first set of K vMF kernels ($\boldsymbol{\mu}^I$) are used for training while the other set is not part of the training pipeline. The network parameters are randomly initialized and trained for 15 epochs. In Stage 2 of the training, the network is initialized with the weights learned in Stage 1. Additionally, the remaining set of K vMF kernels ($\boldsymbol{\mu}^{II}$) are randomly initialized. The dissimilarity loss is constrained to be above -0.02, meaning its gradient does not propagate when the loss falls below this threshold. Empirically, this chosen threshold yields better results; without it, the tool channels fail to capture meaningful information. Such thresholding ensures that the vMF kernels associated with tools are not excessively pushed away from the non-tool vMF kernels. The network is trained for 10 epochs. The hyperparameters used for training are given in Sec. A.8.

3.2 Stage 1: Learning non-tool representations using vMF kernels

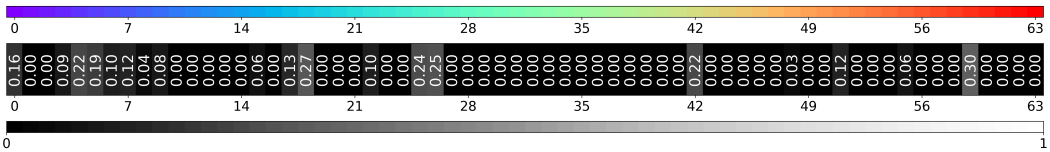
After Stage 1 of the training process, the relative angles of the vMF kernels ($\boldsymbol{\mu}^I$), visualized post dimensionality reduction, are shown in Fig. 2a. The kernels are mainly split into two clusters, one centered near $180°$ and the other centered near $330°$. This is a result of the combined effect of the losses employed. While reconstruction loss forces the kernels to learn semantic information and vMF loss forces them to align towards the vMF clusters in the latent representation $\mathbf{z}$, the dissimilarity loss reduces the similarity between the kernels. As a result, only a subset of kernels in $\boldsymbol{\mu}^I$ learn meaningful information while the others are pushed away from these kernels.



(a) Relative orientation of kernels in $\boldsymbol{\mu}^I$ after Stage 1.



(b) Relative orientation of kernels in $\boldsymbol{\mu}$ after Stage 2.



(c) Average entropy of channels in disentangled latent representations $\mathbf{z}_{vMF}$ after Stage 2.

Figure 2: Orientation of the vMF kernels and the entropy of the corresponding channels of $\mathbf{z}_{vMF}$.

3.3 Stage 2: Learning tool representations using vMF kernels

In Stage 2, vMF loss updates the weights of a vMF kernel using the feature vectors with a higher likelihood of belonging to the cluster represented by that kernel. As $\boldsymbol{\mu}^I$ learns non-tool information, it is crucial to prevent $\boldsymbol{\mu}^{II}$ from learning the same to achieve proper disentanglement. Dissimilarity loss is employed to enforce this separation, which reduces the similarity between the non-tool and tool kernels. Hence, the kernels in $\boldsymbol{\mu}^{II}$ are expected to model the vMF clusters belonging to tool feature vectors. The relative orientation of the vMF kernels upon training is shown in Fig. 2b. There is a visual separation between the two sets of vMF kernels because the dissimilarity loss pushes

away the second set of vMF kernels. Entropy calculated for each channel of $\mathbf{z}_{vMF}$ averaged across the validation dataset is shown in Fig. 2c.

3.4 Limitations of vMF loss in learning disentangled representations

Based on the hypothesis, to have vMF kernels $\boldsymbol{\mu}^{II}$ learn tool information, the feature vectors of tools must be oriented away from the non-tool feature vectors. Conversely, the feature vectors of tools and non-tools are highly similar. To verify and support this, the trained $\boldsymbol{\mu}^{I}$ and randomly initialized $\boldsymbol{\mu}^{II}$ kernels are taken, and the cosine similarity between the feature vectors of images with tools and these vMF kernels was computed. All the feature vectors, including those corresponding to the tools, belong to the clusters represented by the non-tool vMF kernels and none to the randomly initialized set of vMF clusters. Considering the working of vMF loss (Sec. A.5), the tool vMF kernels are not being updated enough using the feature vectors and hence do not learn any tool-specific information. The distribution of feature vectors among the vMF kernels is shown in Sec. A.7. Even when the non-tool vMF kernels are initialized using the vMF mixture model approach as used in the works Liu et al. (2022); Kortylewski et al. (2020), $\boldsymbol{\mu}^{I}$ kernels are closer to the feature vectors than $\boldsymbol{\mu}^{II}$ kernels. Thus, the disentanglement of representations through a two-stage process leads to the second set of vMF kernels not learning the distribution of feature vectors.

3.5 Analyzing effective change in the vMF kernels

Effective change for a given kernel, defined by cosine similarity between the state of kernels after Stage 1 and Stage 2, is shown in Fig. 3. These values indicate the degree of kernel update due to training. The average effective change for the non-tool kernels $(\boldsymbol{\mu}^{I})$ is 0.91, whereas for the tool kernels $(\boldsymbol{\mu}^{II})$ it is 0.49. Smaller area tools occupy in these images in comparison to the background is a possible reason for such minimal updates between stages and the observations mentioned in Sec. 3.4. Moreover, the higher change observed in tool vMF kernels is due to the dissimilarity loss pushing these kernels away from non-tool vMF kernels but not them learning the tool representations. In support of this, few channels in $\mathbf{z}^{II}_{vMF}$ have non-zero entropy (Fig. 2c). This is also due to the dynamics of the losses, causing few kernels in the set $\boldsymbol{\mu}^{II}$ to learn some information from $\mathbf{z}$.

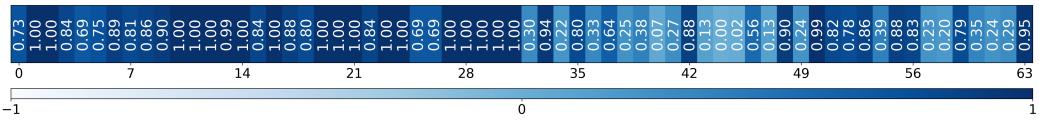


Figure 3: Effective change for each kernel from Stage 1 to Stage 2.

3.6 Reconstruction from selected channels of latent representation

Although the proposed method has not successfully achieved complete disentanglement concerning the tool and non-tool representations, partial disentanglement has been observed between brighter and darker regions of the images. Analysis of the disentangled latent representation, $\mathbf{z}_{vMF}$, reveals that specific channels selectively encode information from bright regions. Reconstructing images using only these channels results in outputs that preserve only the bright pixel regions of the image. This demonstrates the potential of the learned representations to isolate and reconstruct specific information in compressed representation. Detailed results are provided in the Sec. A.11.

4 Conclusion

This work has experimented with a two-stage training process to learn the disentangled representations of surgical images. It provides concise insights on the working of vMF loss and analyzes its limitations in such a two-stage training process. Additionally, a demonstration of the use of disentanglement to generate images with specific information is shown. Although it aimed for object-level disentanglement, it has achieved partial pixel-intensity-based disentanglement, highlighting challenges in achieving true semantic separation of the compressed representations when objects occupy smaller areas and without external supervision.

REFERENCES

Mischa Dombrowski, Hadrien Reynaud, Matthew Baugh, and Bernhard Kainz. Foreground-background separation through concept distillation from generative image foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 988–998, 2023.

Ravi Kakaiya, Rakshith Sathish, Debdoot Sheet, and Ramananthan Sethuraman. Exploiting richness of learned compressed representation of images for semantic segmentation. In *2023 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 482–485. IEEE, 2023.

Adam Kortylewski, Ju He, Qing Liu, and Alan L Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8940–8949, 2020.

Moez Krichen. Convolutional neural networks: A survey. *Computers*, 12(8):151, 2023.

Xiao Liu, Spyridon Thermos, Pedro Sanchez, Alison Q O'Neil, and Sotirios A Tsaftaris. vmfnet: Compositionality meets domain-generalised segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 704–714. Springer, 2022.

Aditya Raj, Rakshith Sathish, Tandra Sarkar, Ramanathan Sethuraman, and Debdoot Sheet. Designing deep neural high-density compression engines for radiology images. *Circuits, Systems, and Signal Processing*, 42(2):643–682, 2023.

Snehal Singh Tomar and AN Rajagopalan. Latents2segments: Disentangling the latent space of generative models for semantic segmentation of face images. *arXiv preprint arXiv:2207.01871*, 2022.

Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5104–5113, 2020.

A APPENDIX

A.1 VON-MISES-FISHER (VMF) DISTRIBUTION

von-Mises-Fisher (vMF) distributions are probability distributions commonly used in statistics and directional data analysis. They are primarily used to model data points distributed on the surface of a unit hypersphere in an n-dimensional space. These distributions are beneficial when dealing with directional data. Mathematically, the probability density function (PDF) of a vMF distribution for an input $\boldsymbol{x}$ is defined as:

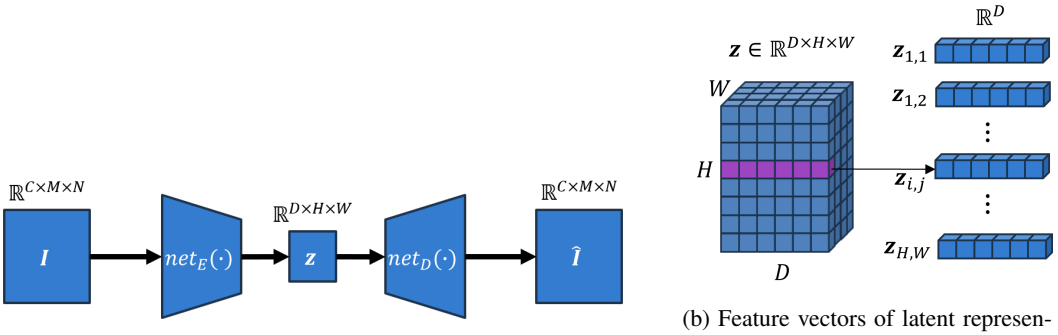
$$f(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\kappa}) = C_p(\boldsymbol{\kappa})^{-1} exp(\boldsymbol{\kappa}\boldsymbol{\mu}^T\boldsymbol{x}) \; s.t. \; \|\boldsymbol{x}\| = 1, \; \|\boldsymbol{\mu}\| = 1, \; \boldsymbol{\kappa} \geq 0 \tag{2}$$

where $C_p(\boldsymbol{\kappa})$ is a normalization constant parameterized by kappa $\boldsymbol{\kappa}$, and mean $\boldsymbol{\mu}$ of the distribution. The parameter $\boldsymbol{\mu}$ defines the average or the expected direction in which the data points are clustered, and the parameter $\boldsymbol{\kappa}$ defines the spread or tightness of the distribution around the average direction specified by $\boldsymbol{\mu}$.

The use of vMF distributions in this work is directly inspired by the work Liu et al. (2022). This work aims to learn the compositional components of two classes, namely the surgical tools and the background. Previous works (Kortylewski et al., 2020) have proven that vMF distributions are crucial in learning compositional components, hence justifying the use of vMF distributions in learning disentangled representations of compositional components in surgical images.

A.2 FEATURE VECTORS

Consider an autoencoder architecture shown in Fig. 4a. It takes an image $\boldsymbol{I} \in \mathbb{R}^{C \times M \times N}$ as input, where C is the number of channels, M and N are the height and width of the image, respectively. The latent representation $\mathbf{z} \in \mathbb{R}^{D \times H \times W}$, is obtained upon passing the image $\boldsymbol{I}$ through the encoder $net_E(\cdot)$, where D is the number of channels, H and W are the height and width respectively.



(a) Autoencoder architecture.

(b) Feature vectors of latent representation.

Figure 4: Schematic represention of the feature vectors obtained from latent representations of encoded input images.

The latent representation $\mathbf{z} \in \mathbb{R}^{D \times H \times W}$ is schematically illustrated as a tensor in Fig. 4b. A feature vector $\mathbf{z}_{(i,j)} \in \mathbb{R}^{D \times 1}$ is extracted from a specific lattice position (i, j) spanning all D channels of the latent representation. Consequently, the latent representation consists of HW feature vectors.

A.3 DISENTANGLEMENT BLOCK

Let $\mathbf{z} \in \mathbb{R}^{D \times H \times W}$ be the latent representation obtained from the encoder network $net_E(\cdot)$. The latent representation is convolved with the vMF kernels parameterized by $\boldsymbol{\mu} = \{\boldsymbol{\mu}^I, \boldsymbol{\mu}^{II}\}$ to obtain the disentangled representation $\mathbf{z}_{vMF}$. In particular, we define $\boldsymbol{\mu}^I = \{\boldsymbol{\mu}_i^I \in \mathbb{R}^{1 \times D} \; \forall \; i \in \{0, 1, \cdots, K-1\}\}$ and $\boldsymbol{\mu}^{II} = \{\boldsymbol{\mu}_i^{II} \in \mathbb{R}^{1 \times D} \; \forall \; i \in \{K, K+1, \cdots, 2K-1\}\}$. Essentially, there are $2K$ vMF kernels that are used for convolving with the latent representation $\mathbf{z}$ to obtain $\mathbf{z}_a \in \mathbb{R}^{2K \times H \times W}$.

$$\mathbf{z}_{a(k,i,j)} = \boldsymbol{\mu}_k^T \mathbf{z}_{i,j} \; \|\mathbf{z}\| = 1, \; \|\boldsymbol{\mu}\| = 1 \tag{3}$$

The exponential operation is applied on the tensor $\mathbf{z}_a$ to obtain $\mathbf{z}_{ad} \in \mathbb{R}^{2K \times H \times W}$:

$$\mathbf{z}_{ad(k,i,j)} = \frac{C_p(\boldsymbol{\kappa})^{-1}}{C_p(\boldsymbol{\kappa})^{-1}} exp(\boldsymbol{\kappa} \cdot \mathbf{z}_{a(k,i,j)}) = C_p(\boldsymbol{\kappa})^{-1} P(\mathbf{z}_{i,j} \; ; \; \boldsymbol{\mu}_k, \boldsymbol{\kappa}) \tag{4}$$

The disentangled representation is computed by normalizing the tensor $\mathbf{z}_{ad}$ across all the channels. This operation is represented by the literal $N(\cdot)$ in Fig. 1. The normalization operation is represented as follows:

$$\mathbf{z}_{vMF(k,i,j)} = \frac{C_p(\boldsymbol{\kappa})^{-1} P(\mathbf{z}_{i,j} \; ; \; \boldsymbol{\mu}_k, \boldsymbol{\kappa})}{\sum_{k=1}^{2K} C_p(\boldsymbol{\kappa})^{-1} P(\mathbf{z}_{i,j} \; ; \; \boldsymbol{\mu}_k, \boldsymbol{\kappa})} = \frac{\mathbf{z}_{ad(k,i,j)}}{\sum_{k=1}^{2K} \mathbf{z}_{ad(k,i,j)}} \tag{5}$$

A.4 COMPOSITION OPERATION

After disentangling the compositional components into individual channels of the tensor $\mathbf{z}_{vMF}$ having $2K$ channels, it is transformed into a tensor with D channels to be given to the decoder network $net_D(\cdot)$. This is necessary to ensure that the number of channels in the encoder's output and in the decoder's input is the same. To facilitate this, Liu et al. (2022) have proposed a compose operation that performs this transformation. The compose operation is an affine transform that takes in the normalized likelihood tensor $\mathbf{z}_{vMF} \in \mathbb{R}^{2K \times H \times W}$ and the set of cluster centers $\boldsymbol{\mu} \in \mathbb{R}^{2K \times D \times 1}$ to produce the composed version $\tilde{\mathbf{z}} \in \mathbb{R}^{D \times H \times W}$. This composed version $\tilde{\mathbf{z}}$ is given to the decoder for reconstruction purposes. This operation is represented as

$$\tilde{\mathbf{z}} = \mathbf{z}_{vMF} \boldsymbol{\mu} \tag{6}$$

A.5 vMF LOSS

The loss formulation is given in Sec. 2.2. The vMF loss is used to learn the mean of the feature vector clusters. The loss can be understood as follows: for all the feature vectors, compute the cosine similarity between that feature vector and all the vMF kernels. Choose the maximum cosine similarity obtained for each feature vector across all the vMF kernels and compute the average cosine similarity to obtain vMF loss. The goal of this loss is to increase the likelihood of the feature vectors under the current vMF clusters parametrized by $\boldsymbol{\mu} \in \mathbb{R}^{2K \times D}$ where $2K$ is the number of vMF clusters, and D is the dimension of the feature vectors.

A.6 DATASET DESCRIPTION

The proposed two stage approach is tested on the Cholec80 dataset containing videos of 80 cholecystectomy surgical videos (Twinanda et al., 2016). Each video is sampled at one frame per second and contains the corresponding images. The annotation for each image indicates the tools present in it.

Among these 80 videos, a random set of 60 videos is chosen for training and the remaining videos are used for validation. These two sets are further divided into images with and without tools. The images without tools from the training set and validation set are used for Stage 1 of the proposed method, and the images with tools from the training set and validation set are used for Stage 2 of the proposed method.

A.7 PERCENTAGE OF FEATURE VECTORS BELONGING TO EACH CLUSTER AFTER STAGE 1

Sec. 3.4 describes that the feature vectors of non-tools are highly similar to the feature vectors of tools. To prove this, the trained set of vMF kernels $\boldsymbol{\mu}^I$ from Stage 1 and the randomly initialized set of vMF kernels $\boldsymbol{\mu}^{II}$ are combined. Cosine similarity between the feature vectors of images with tools and these 64 vMF kernels has been computed. Upon analyzing the cosine similarity, it is observed that all the obtained feature vectors are distributed among the clusters represented by the kernels shown in Tab. 1. This shows that the parameters of randomly initialized kernels $\boldsymbol{\mu}^{II}$ are not updated through the vMF loss. They are updated through the gradients from the other two losses, namely the reconstruction loss and the dissimilarity loss.

Table 1: Percentage of feature vectors belonging to the cluster represented by kernels after Stage 1 (0-based indexing).

Kernel	**Value**
Kernel 1	$1.82 \times 10^{-5}\%$
Kernel 2	$6.09 \times 10^{-6}\%$
Kernel 10	21.15%
Kernel 11	$7.94 \times 10^{-3}\%$
Kernel 14	73%
Kernel 19	$4.9 \times 10^{-1}\%$
Kernel 21	$4 \times 10^{-1}\%$
Kernel 28	3.52%
Kernel 29	$6.07 \times 10^{-3}\%$
Kernel 31	1.03%

A.8 HYPERPARAMETERS

The hyperparameters used for training the proposed architecture are given below.

Table 2: Hyperparameters used for training (common for Stage 1 and Stage 2).

Parameter	Value
K	32
Learning rate	0.001
Batch size	32
Learning rate scheduler type	Step
Scheduler step	10
Scheduler rate	0.9
$\boldsymbol{\kappa}$	30
$\boldsymbol{\alpha}_1$	3
$\boldsymbol{\beta}_1$	0.05
$\boldsymbol{\alpha}_2$	1
$\boldsymbol{\beta}_2$	0.001

The hyperparameters for weights of the losses are chosen by analyzing the feature disentanglement visually for multiple combinations of weights. The visual analysis focused on two aspects of the disentangled representations: amount of diversity in the disentangled features and the number of channels in the disentangled representation that contain some information. The lack of labeled ground truth data for the compositional components in the images is the key reason for following a visual approach in choosing hyperparameters for this work.

A.9 Disentangled latent representation of images with no tools

The disentanglement achieved after Stage 1 is shown for a sampled set of images from the validation dataset. For these images, each channel in $\mathbf{z}_{vMF}$ is displayed in Fig. 5. 0-based indexing is used to number the channels, and the index for the first channel of each row is written to the left of it. The border color for each channel image is according to the color map generated for Fig. 2c. The intensity scale chosen for the channels' values is [0, 1].

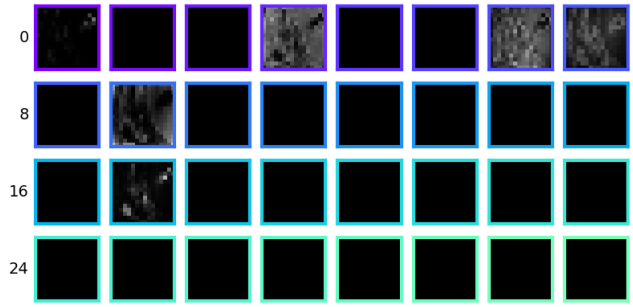
A.10 Disentangled latent representation of images with tools

To visualize the disentanglement achieved after Stage 2, one image from each class is sampled from the validation data, and each channel in $\mathbf{z}_{vMF}$ is displayed for all these images in Fig. 6. The sampled images are shown in the first column of Fig. 7. 0-based indexing is used to number the channels, and the index for the first channel of each row is written to the left of it. The border color for each channel image is according to the color map generated for Fig. 2c. The intensity scale chosen for the values in channels from 0 to 31 is [0, 1], while for the channels with an index from 32 to 63, it is [0, 0.2]. This is chosen to enhance the visibility of the captured features.

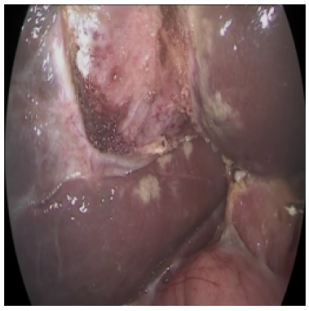
A.11 Progressive reconstruction

Progressive reconstruction is carried out by grouping the disentangled channels into four categories. Category A includes channels that primarily capture bright regions, while Category B consists of channels that contain a mix of bright and dark regions. Category C includes more channels with prominent dark regions, and Category D comprises all latent channels from $\mathbf{z}_{vMF}$.

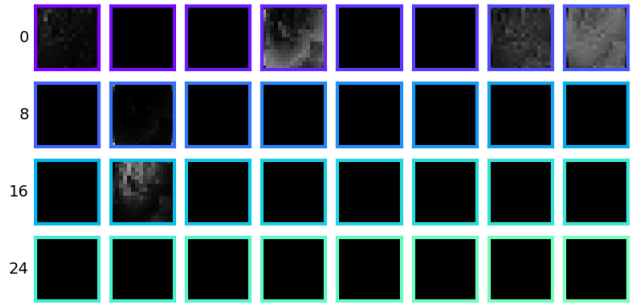
(a) Sample 1



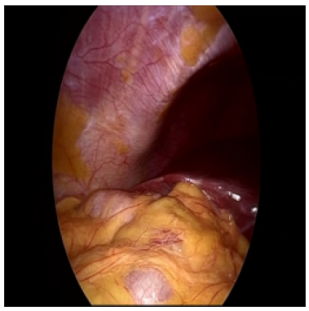
(b) Sample 1 compressed representations.



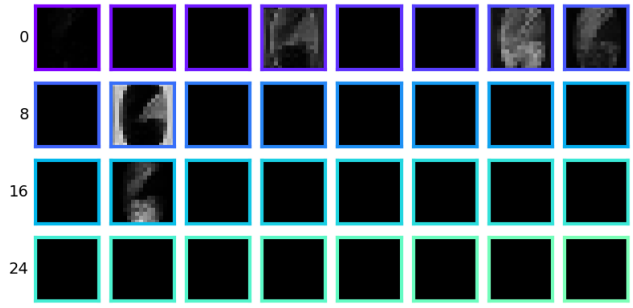
(c) Sample 2



(d) Sample 2 compressed representations.

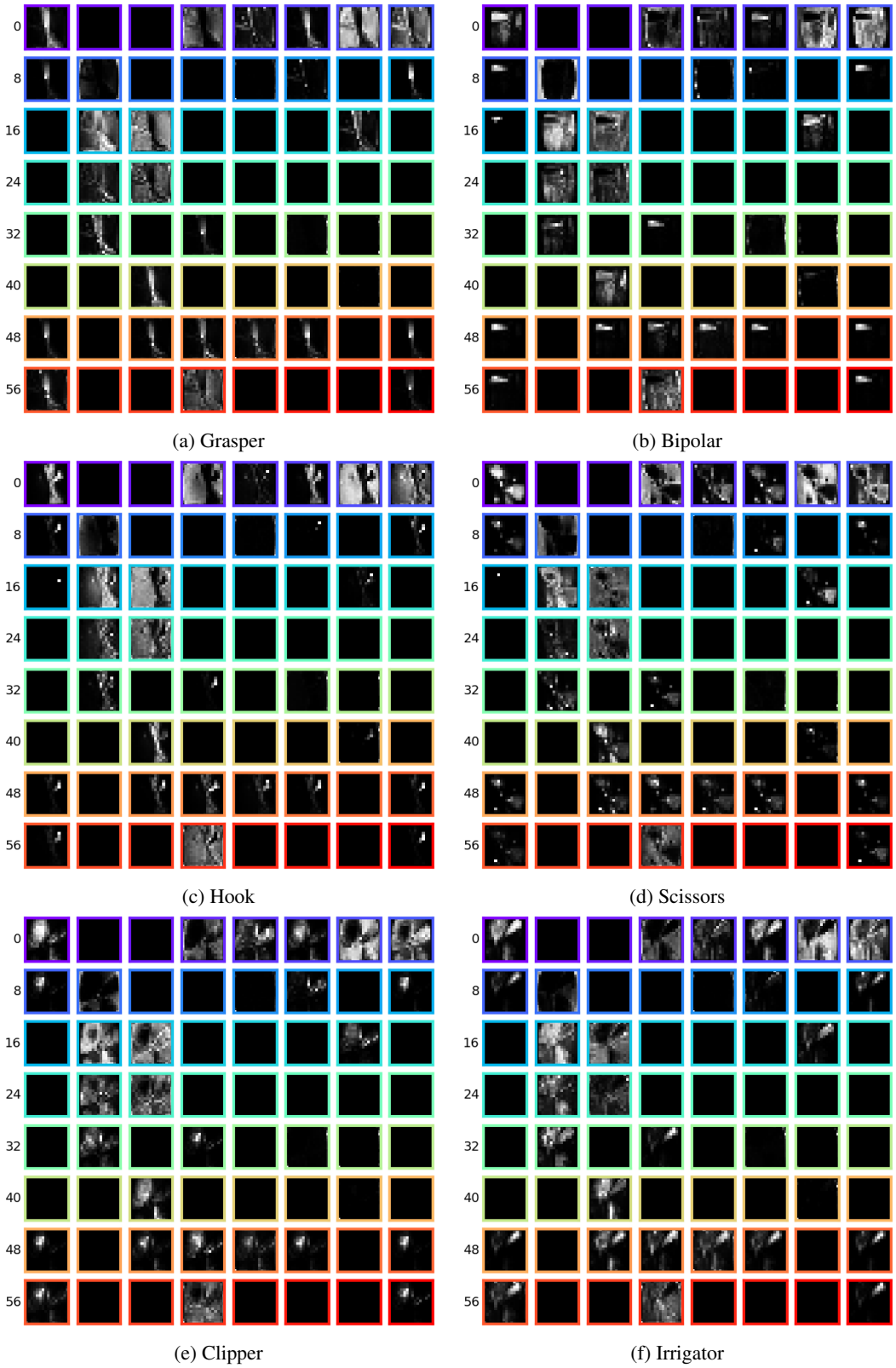


(e) Sample 3



(f) Sample 2 compressed representations.

Figure 5: Sample images with corresponding $\mathbf{z}_{vMF}$ representing the disentanglement achieved after Stage 1.



(a) Grasper (b) Bipolar

(c) Hook (d) Scissors

(e) Clipper (f) Irrigator

Figure 6: $\mathbf{z}_{vMF}$ representing the disentanglement achieved after Stage 2 for various tool classes for Sample images in Fig. 7.

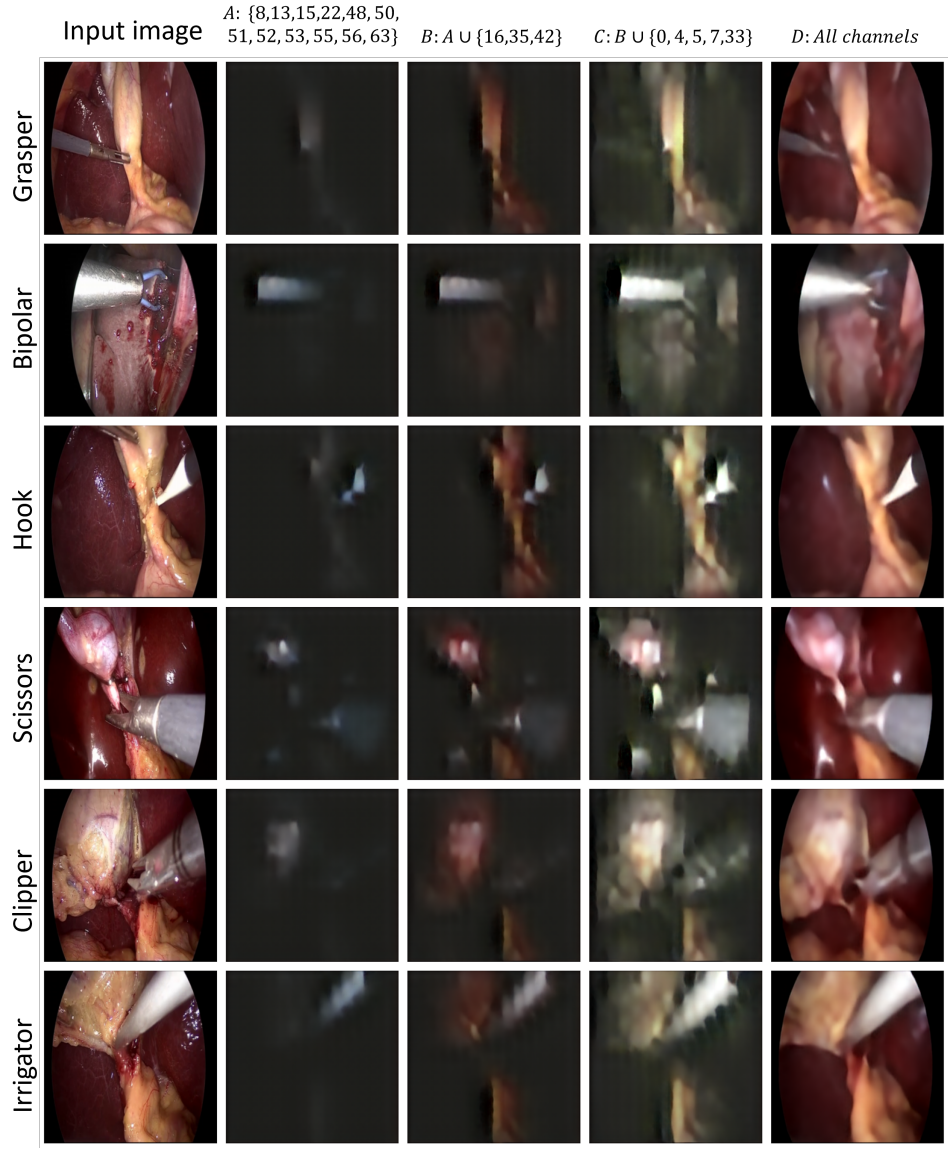


Figure 7: Progressive reconstruction: An overview of reconstructed images when using the specific channels from the $\mathbf{z}_{vMF}$ for reconstruction. The reconstructed images from the initial select channels prove partial pixel-intensity-based disentanglement of the compressed features argument, and this is consistent across multiple samples containing various tools.