In this section, we first show that motion operation and convolution operation can be expressed as matrix operations (Lemma 1). Then we will proceed to prove the existence of a linear transformation that maps the feature map of key frame to the feature map of non-key frames based on the motion information (Theorem 2). Finally, we will show that the error term of residual map will not explode after a sequence of convolution operations, guaranteeing that flowed feature maps enjoys only a negligible error from the convolution feature map generated from the non-key frame (Theorem 3).

Lemma 1. *Motion operation $\mathcal{M}_{\mathcal{A}\to\mathcal{B}}$ and convolution operation C can be expressed as matrix operations, thus linear operations.*

Proof: For an arbitrary image $\mathcal{A}$, we can unroll into vector

$$\mathcal{V} = \{\mathcal{A}_{00}, \mathcal{A}_{01}, \ldots, \mathcal{A}_{0d}, \mathcal{A}_{10}, \mathcal{A}_{11}, \ldots, \mathcal{A}_{1d}, \ldots, \mathcal{A}_{d0}, \mathcal{A}_{d1}, \ldots, \mathcal{A}_{dd}\}$$

where $\mathcal{A}_{ij}$ stands for the element of $\mathcal{A}$ at row i and column j and generate permutation matrix $\mathcal{T}$ by the pixel repositioning relationship given by flow field $M_{i\to k}$. Then, we can multiply the vector and the permutation matrix in ordinary way as a shuffle of elements:

$$\mathcal{V}' = \mathcal{V} * \mathcal{M}_{\mathcal{A}\to\mathcal{B}}$$

Finally, we can reshape the vector back to matrix

$$\mathcal{A}' = \{\mathcal{V}'_0, \ldots, \mathcal{V}'_{d-1}; \mathcal{V}'_d, \ldots, \mathcal{V}'_{2d-1}; \ldots; \mathcal{V}'_{(d-1)*d}, \ldots, \mathcal{V}'_{d*d}\}$$

where $\mathcal{V}'_i$ stands for the ith element of $\mathcal{V}'$. Following this "unroll-matrix-vector-multiply-reshape" pattern, we can convert the motion operation to a linear operation. Similarly, we can also convert the convolution operation $\mathcal{C}$ into a linear operation. ■

Based on the linearity of motion operation and convolution operation, we can show the existence of linear transformation between feature maps of key frame $\mathcal{A}$ and the feature map of non-key frame $\mathcal{B}$ as follows:

Theorem 2. *Given a convolution operation $\mathbb{C}$, $\forall$ frame $\mathcal{A}$ and $\mathcal{B}$, as well as the corresponding feature maps $\mathcal{A}'$ and $\mathcal{B}'$, $\exists$ a linear transformation $\mathcal{T} = \mathcal{C}^{-1} \cdot \mathcal{M}_{\mathcal{A}\to\mathcal{B}} \cdot \mathcal{C}$, such that $\mathcal{B}' = \mathcal{A}' \cdot \mathcal{T} + \delta'$, where $\delta' = \delta C$, where $\mathcal{M}_{\mathcal{A}\to\mathcal{B}}$ and δ are motion and error information extracted from motion vector and residual map respectively.*

This theorem shows that the motion information between two frames can be used to generate a linear transformation between the corresponding two feature maps. Intuitively, the convolution operation can be unrolled into a linear operation, thus the composition of motion between frames and convolution still enjoys the linearity. For simplicity, we cover the case when a single convolution operation is applied.

Proof: Extracting from compressed video the motion information $\mathcal{M}_{\mathcal{A}\to\mathcal{B}}$ in motion vector and the error information δ from the residual map, we have

$$\mathcal{B} = \mathcal{A}\mathcal{M}_{\mathcal{A}\to\mathcal{B}} + \delta \tag{1}$$

Following existing works [1], we represents 2D convolution operation as matrix multiplication:

$$\mathcal{A}' = \mathcal{A}\mathcal{C} \tag{2}$$

where $\mathcal{A}'$ stand for the feature map after single layer convolution. Similarly we have:

$$\begin{aligned}
\mathcal{B}' &= \mathcal{B}\mathcal{C} \\
&= (\mathcal{A}\mathcal{M}_{\mathcal{A}\to\mathcal{B}} + \delta)\mathcal{C} \\
&= \mathcal{A}\mathcal{M}_{\mathcal{A}\to\mathcal{B}}\mathcal{C} + \delta\mathcal{C} \\
&= \mathcal{A}'\mathcal{C}^{-1}\mathcal{M}_{\mathcal{A}\to\mathcal{B}}\mathcal{C} + \delta\mathcal{C} \\
&= \mathcal{A}'\mathcal{T}' + \delta'
\end{aligned} \tag{3}$$

■

Remark: This proof for single convolution layer can be generalized to multiple convolution layers since the composition of multiple linear operations is still a linear operation. Specifically, given a sequence of convolution operations $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n$, we can generate the corresponding matrix operations $C_1, C_2, ..., C_n$ following Lemma 1. The composition of convolution operations $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_n$ can be treated as matrix multiplication $C_1 C_2 \cdots C_n$, which is still a linear operation. Substituting $\mathcal{C}$ in equation 3 with $C_1 C_2 \cdots C_n$, we can show the existence of linear transformation when multiple convolution layers exist.

In feature flow, we exploit the motion information for computation reuse, while not utilizing the error information captured by the residual map, in order to reduce computation. In the following theorem, we show that the error information does not magnify after convolution operations, guaranteeing a negligible error of the flowed feature map. To quantify the convolution weights, we assume the unit normality at the convolution filter level, *i.e.*, $d * Var[C_{kj}] \sim N(0, 1)$, following current theoretical analysis [2, 3] on convolution weights, where (k, j) is the location of weights and d is the total number of weights. In addition, we assume that the error information is a white noise, *i.e.*, $\delta \sim N(0, \sigma^2)$, since the patterns across frames have been captured explicitly by the motion information. Intuitively, this error information is independent from the convolution weights, since the error information from testing frames not used in the training procedure.

Theorem 3. Given a convolution operation C with unit normality and an error information $\delta \sim \mathcal{N}(0, \sigma^2)$, the error information δ' after convolution operation enjoys convolution-invariance, *i.e.*, $\delta' = \delta C \sim \mathcal{N}(0, \sigma^2)$.

Proof: For arbitrary pixel δ'_{ij} at location (i, j), we have

$$\begin{aligned} E[\delta'_{ij}] &= E[\sum_k \delta_{jk} * \mathcal{C}_{kj}] \\ &= \sum_k E[\delta_{jk} * \mathcal{C}_{kj}] \\ &= \sum_k E[\delta_{jk}] * E[\mathcal{C}_{kj}] \\ &= \sum_k \delta_{jk} * 0 \\ &= 0 \end{aligned} \tag{4}$$

Here, the first equality follows the definition of convolution operation and the second equality comes from the property of expectation. The third equality holds since error term δ_{jk} and the convolution weights C_{kj} is independent and the forth equality comes from the unit normality of convolution weights.

To quantify the variance, we have

$$\begin{aligned} Var[\delta'_{ij}] &= Var[\sum_k \delta_{ik} * \mathcal{C}_{kj}] \\ &= \sum_k Var[\delta_{ik} * \mathcal{C}_{kj}] \\ &= \sum_k Var[\delta_{ik}] * Var[\mathcal{C}_{kj}] \\ &= \sigma^2 * \sum_k Var[\mathcal{C}_{kj}] \\ &= \sigma^2 * d * Var[\mathcal{C}_{kj}] \\ &= \sigma^2 \end{aligned} \tag{5}$$

Here, the second and third equation holds due to the independence between convolution weights and error terms. ■

References

[1] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *ArXiv*, abs/1603.07285, 2016.

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.