* In the early stages of training, the model's weights are initialized randomly, and the changes in importance are usually significant as the model adapts to the data in a random way.
* Some features in the data may be inherently important for the task and remain relatively stable throughout training. The model may need to adapt continuously to different patterns and may change its focus frequently.
* Regularization: The use of regularization techniques (e.g., L1 or L2 regularization) can encourage the model to maintain feature importance stability by penalizing large weight changes. This can result in more consistent feature importance over epochs.
* Data Variation: If the dataset is diverse and contains various examples, the model might need to adapt to different subsets of features for different samples. This can lead to changes in feature importance.
* Network Architecture: The choice of network architecture can also influence how feature importance evolves during training. Deeper networks may exhibit more complex and dynamic changes in feature importance.
* Hyperparameters: Hyperparameters like learning rate, batch size, and optimization algorithm can impact the convergence speed and stability of feature importance. Poorly chosen hyperparameters may result in erratic changes in feature importance.
* Semantic Information: Kernels of different sizes capture different levels of semantic information. Smaller kernels like 1x1 are often used to capture pointwise interactions and can be thought of as controlling the depth dimension of the feature space. Larger kernels like 3x3 or 5x5 capture more complex spatial patterns.
* Hierarchical Features: Inception-style architectures leverage the idea of capturing hierarchical features. Kernels of different sizes help the model learn features at different scales. Smaller kernels might capture fine-grained details, while larger kernels might capture broader contextual information.
* Feature Combinations: The different kernel sizes allow the model to explore various combinations of features. Smaller kernels can focus on specific feature combinations, while larger kernels can integrate information from a wider region.
* Task-specific Patterns: The choice of kernel sizes often depends on the specific patterns relevant to the task. Some tasks may benefit from fine-grained details (small kernels), while others may require a broader context (large kernels).
* Training Dynamics: During training, the model learns to assign importance to kernels based on their ability to contribute to minimizing the loss function. This importance score assignment can vary based on the task's requirements and the patterns present in the data.
* Regularization: Some kernel sizes may be more heavily regularized than others, affecting their importance values. Regularization techniques like dropout or weight decay can influence the behavior of individual kernels.

Three new possible directions:

* Finish all planned experiments but this time try to find the correlation between weight change/all weight change in the layer after each epoch, with the change in its importance scores after this same epoch
* Show that importance scores on kernels of different sizes will be different on average no matter how we randomly initialize the model weights.