# ConfusionFlow: Visualizing Neural Network Confusion Across Epochs

Martin Ennemoser*
Johannes Kepler University Linz

Peter Ruch†
Johannes Kepler University Linz

Holger Stitz‡
Johannes Kepler University Linz

Hendrik Strobelt§
IBM Research

Marc Streit¶
Johannes Kepler University Linz

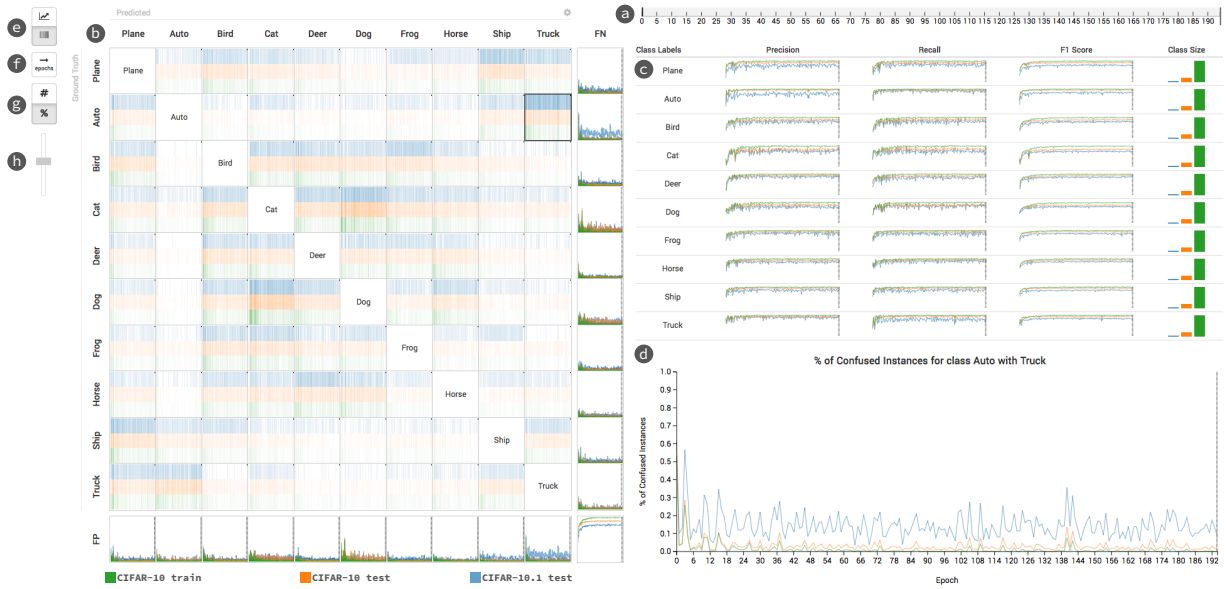


Figure 1: ConfusionFlow consists of three main parts: the timeline (a), the confusion flow visualization (b), and a detail view (c). We show the train set ■ and test set ■ from CIFAR-10 [4] and a recently proposed new test set ■ from CIFAR-10.1 [6]. The line chart (c) shows that the relative number of misclassified images for the selected classes *Auto* and *Truck* deviates notably between the original and the new test set, while for the remaining classes the new test set is able to match the distribution of CIFAR-10 (b).

## Abstract

In order to monitor the learning process and track model quality, training of a neural network on a classification task is usually accompanied by accuracy and loss curves and the performance of the final model is summarized using a confusion matrix. However, showing the final result only completely disregards the change (flow) of the model confusion across epochs of the learning process.

We propose ConfusionFlow, a generalization of the confusion matrix concept over time that enables the user to uncover the learning dynamics of the neural network model.

As a first step towards a more informed design process for network architectures and selection of an optimization procedure and its hyperparameters, ConfusionFlow allows for interactive, comparative exploration of model confusion over the network's learning process.

**Keywords:** Confusion matrix, evolution, multi-class classifier.

## 1 Introduction

Neural networks are a model family that recently gained a lot of popularity due to their exceptional performance on many different prediction problems. However, finding a model that performs well for a specific task is challenging.

A neural network model employs a specific network architecture such as *VGG* or *ResNet* for image models. In order to obtain a final model, the network parameters are optimized iteratively over many epochs using a model selection procedure, an optimization method such as *Adam* or *SGD*, with different hyperparameters (e.g., learning rate, batch size, momentum). The combination of a specific network architecture, the used optimizer with its hyperparameters—referred to as *run configuration*—can heavily impact the final results.

During training, the model increasingly learns to differentiate between classes. The untrained network has initially low accuracy which leads to high error rates when instances are classified. Detecting correlations between the class-wide mispredictions is commonly performed using a confusion matrix. While the model confusion is often only evaluated for the final model, the overall model quality is closely monitored during training using line charts with accuracy and loss values for each epoch. Especially in classification, the quality of the final model is determined by several performance metrics, such as precision and recall, which are carefully monitored during the learning process. Performance curves, such as accuracy and loss value plots, are commonly used visualizations that provide model-wide insight into the training process. To the best of our knowledge, only little attention has been given to the observation of temporal changes on class level.

We propose ConfusionFlow, an interactive visual analysis tool that provides insights into the training process along time *and* classes by combining the confusion matrix and performance curves. ConfusionFlow is able to (1) visualize the classification behavior of a

*e-mail: martin.ennemoser@jku.at
†e-mail: peter.ruch@jku.at
‡e-mail: holger.stitz@jku.at
§e-mail: hendrik.strobelt@ibm.com
¶e-mail: marc.streit@jku.at

single model over multiple epochs and (2) compare the classification behavior of multiple run configurations over time. Further, ConfusionFlow supports analysts to detect mismatches in the underlying data distribution and shows additional details about class confusion over time that usually remain hidden.

## 2 Related Work

The research community has quickly recognized how visual analytics can contribute to the emerging field of deep learning and how visualization tools can aid the interpretability and further understanding of the complex models which are used today [2].

Apart from several custom visualization systems, such as ActiVis [3], which are designed to cover many quality aspects of a learned model, only little work has been done on improving existing visualization techniques that are commonly used to visualize model performance and diagnose errors.

Alsallakh et al. [1] use a confusion matrix based approach to identify misclassified instances and to reason the model's performance.

Current tools lack the possibility to track model confusion across epochs or to provide a fine-grained comparison of the classification behavior of multiple run configurations.

## 3 ConfusionFlow Concept

The ConfusionFlow interface consists of three linked components, as illustrated in Figure 1: (a) the **timeline** for selecting the range of epochs that are used for exploration , (b) the **confusion flow visualization** presenting the confusion of the model over time, (c) per-class performance measures (such as precision or recall), and (d) a **detail view** visualizing the performance curves for the selected class pairing in the confusion flow visualization in greater detail. Each selected run configuration gets a unique color assigned in order to distinguish multiple run configurations in the ConfusionFlow matrix and the detail view.

The ConfusionFlow visualization is an extension of the classic confusion matrix where the *actual* or *ground-truth* class-labels are organized in *rows*, and the *predicted* labels are organized in the columns. Cells of a classic confusion matrix show the number of instances of the *row*-class that were predicted as *column*-class at a specific epoch. In contrast, ConfusionFlow visualizes the confusion for a specific class pairing (cell) by plotting the values within the selected epoch range as line chart or heatmap (Fig. 1e), thus allowing a trend comparison of multiple runs using superimposed lines or stacked heatmaps. The heatmap visualization is the default encoding for cell pairings and encodes the value as brightness with same hue as assigned to the run configuration. Users can increase the contrast to emphasize small values by using the exponential scaling slider (Fig. 1h). We align epochs horizontally to facilitate the comparison across cells along the *predicted* axis and let users rotate the cell content by 90° (Fig. 1f) to compare cells along the *ground-truth* axis. Additionally, users can switch from absolute to relative performance values (Fig. 1g) to compare the performance for different dataset sizes (e.g., training versus test set).

ConfusionFlow currently supports up to around ten classes. Datasets with notably more classes require pre-processing, for instance sub-sampling or other aggregation strategies. As the width of the heatmap bars is driven by the number of epochs, the limited space in a ConfusionFlow cell can lead to sampling problems. Reducing the epoch resolution using interpolation or sub-sampling, however, can potentially obscure interesting patterns or events in the evolution data. Further, ConfusionFlow supports up to four selected run configuration to reserve enough visual space per configuration in the heatmap representation and avoid visual clutter in the line chart representation.

## 4 Implementation

The datasets are generated by logging the ground-truth and predicted class labels of a data instance subset from a Jupyter notebook for every epoch. Small datasets can be logged completely, large datasets require the creation of a representative subset. The data is then imported into ConfusionFlow, a server-client application based on the Caleydo Phovea framework[1]. The server side is written in Python and the client side is written in TypeScript using D3.js. The prototype of ConfusionFlow is available at confusionflow.caleydoapp.org[2].

## 5 Usage Scenario

The evolution of the confusion matrix over time allows users to detect differences in the data distributions between different subsets of the same dataset. Recent work [6] proposes a new test set CIFAR-10.1 for the CIFAR-10 image dataset [4]. Figure 1 illustrates that while the CIFAR-10.1 dataset matches the underlying distribution of the train and test sets for CIFAR-10 well, there seems to be an issue for the pairing *Auto* and *Truck*. The classification error is significantly worse which becomes immediately obvious when looking at the line chart and the outlined heatmap cell (see Figure 1d).

## 6 Conclusion and Future Work

In this poster we presented ConfusionFlow, a novel tool for visualizing and exploring the class confusion of an iteratively trained multi-class classifier.

To incorporate initial user feedback and based on internal discussions, we are currently looking into ways how to represent instance-specific information and how to link it with feature visualizations.

**Instance Specific Information** At its current state, ConfusionFlow does not support the tracking of confusion on an instance-based level over the learning process. However, exploring the learning dynamics of instances could provide valuable additional insight into the underlying dataset and the classification behavior of individual runs. Detecting problematic instances which are misclassified could potentially help to detect labelling errors or outlier instances that might otherwise go by unnoticed.

**Feature Visualization** By visualizing the class confusion over time, ConfusionFlow can reveal interesting events during the learning phase, for instance performance regressions. In the context of convolutional networks, linking these events to changes in the learned representations, utilizing feature visualization techniques [5], could improve the understanding of the models' learning behavior.


## Acknowledgments

This work was supported in part by the Austrian Science Fund (FWF P27975-NBL) and the State of Upper Austria (FFG 851460).


## References


[1] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. The State-of-the-Art of set visualization. In *Computer Graphics Forum*, vol. 35, pp. 234–260. Wiley Online Library, 2016.

[2] F. M. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Trans. Vis. Comput. Graph.*, June 2018.

[3] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Polo Chau. ACTIVIS: Visual exploration of Industry-Scale deep neural network models. *IEEE Trans. Vis. Comput. Graph.*, Aug. 2017.

[4] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[5] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

[6] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv*, 2018.


[1] http://phovea.caleydo.org/

[2] https://confusionflow.caleydoapp.org/