Screenshots from IRVINES' Four Iterations

We present you screenshots of IRVINE made during the four system iterations. The first two iterations were mainly about setting up the systems backend pipelines (e.g. Hypermatrix computations or databases). Hence, not many actual system screenshots are provided here. The third iteration was also much about working on the retraining of our SOM, which is why many screenshots of the "SOM Training Dialog" are provided there. Thus, most of the screenshots, which show the entire system, are provided for the fourth iteration.

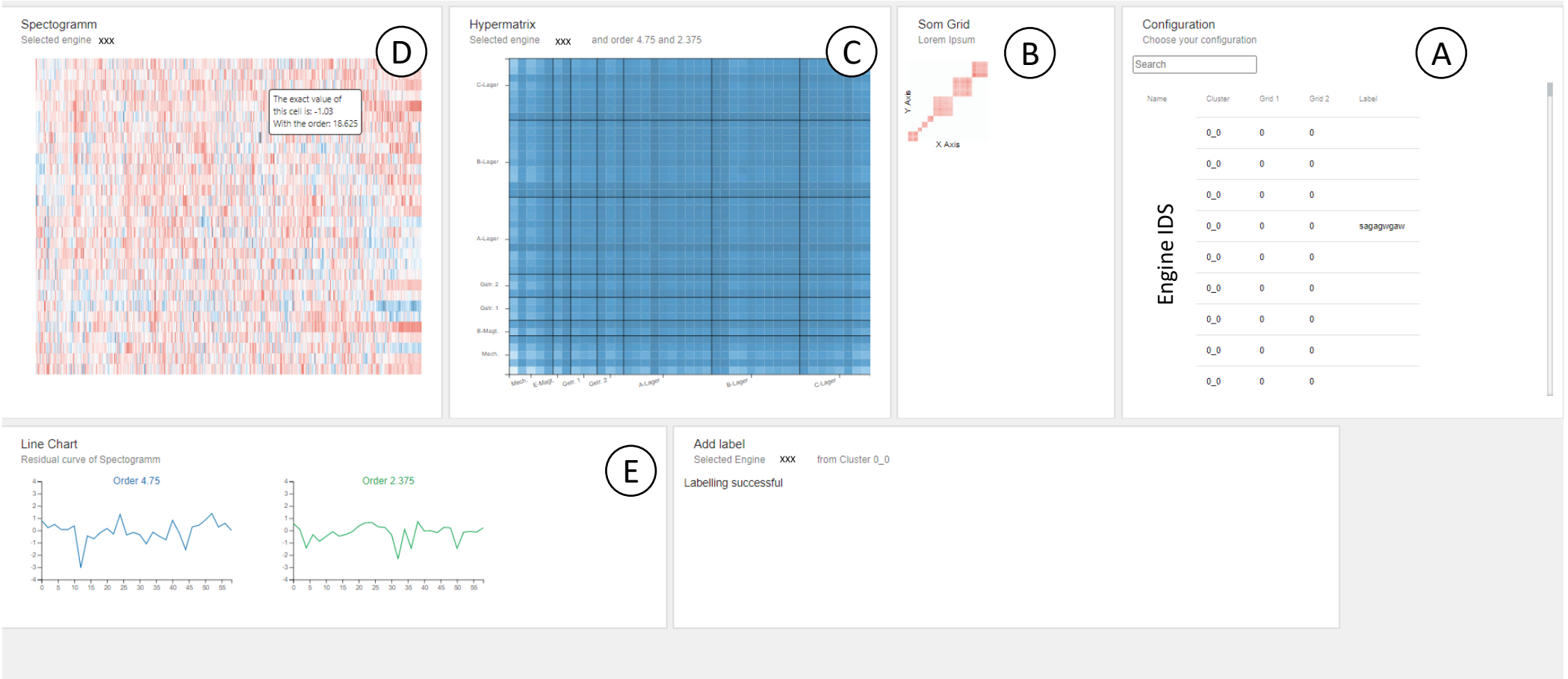
Due to non-disclosure agreements with BMW, we cannot show Engine Ids.

First Scribble of IRVINE

The scribble from the first meeting, where we discussed IRVINE with our lead user and visualization expert. The Idea of showing the spectrogram, the Hypermatrix, and a SOM was already there.

First Running Version of IRVINE

Iteration 1



First example of how prototypical views could look like. We already included data from 53 engines (A). A good color theme was missing and only one cluster for testing purposes was shown (B). Initial coupling between the Hypermatrix (C) and spectrogram (D) was implemented via showing the order lines (E) of the spectrogram when hovering over the Hypermatrix. Part of this iteration was also to setup the entire backend for the system (e.g. Hypermatrix and SOM computation). In this stage IRVINE did run on local json files.

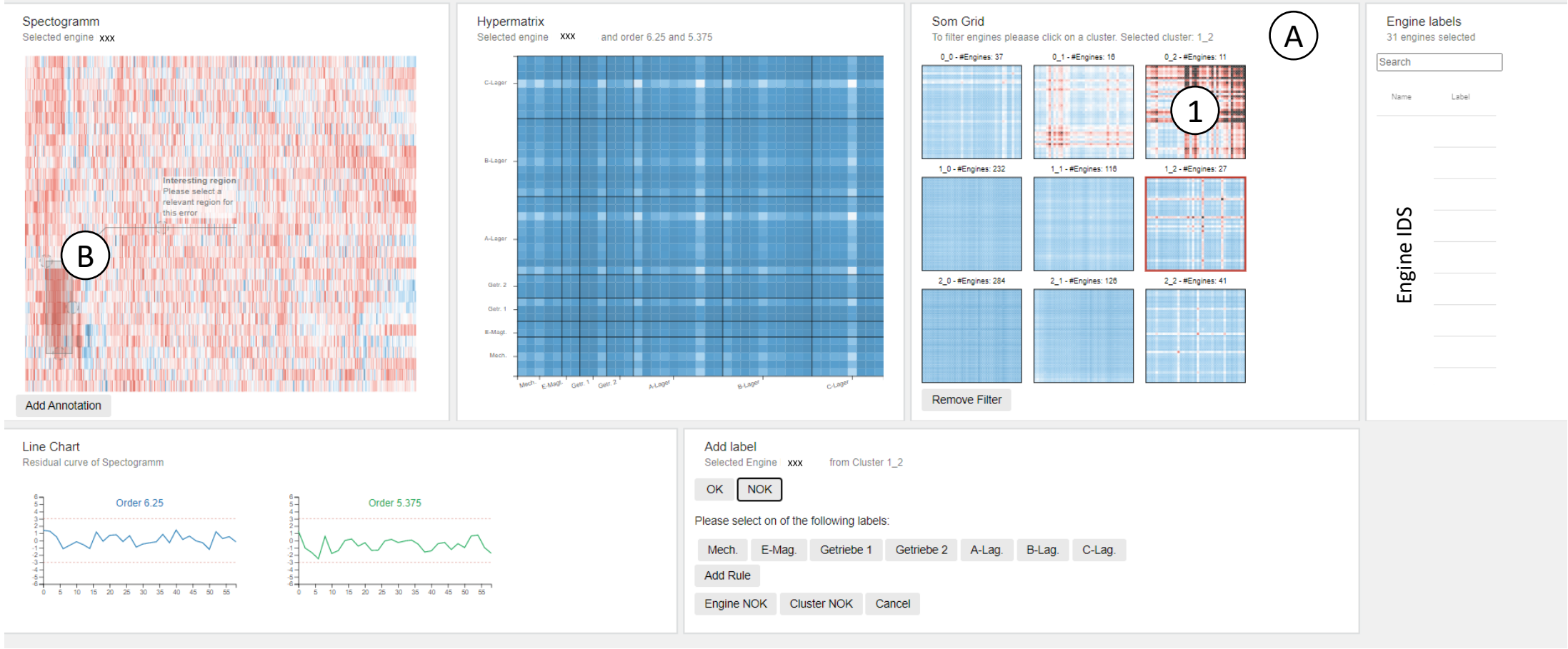
Working on the SOM and Labeling

Iteration 2

Here, we showed how SOMs (A) serve well to cluster similar correlation patterns of Hypermatrices. In this stage, we showed only submatrices of single engine components in the SOM view (A1, A2, A3). Also a first labeling functionality was introduced (B).

Implemented Basic Annotation

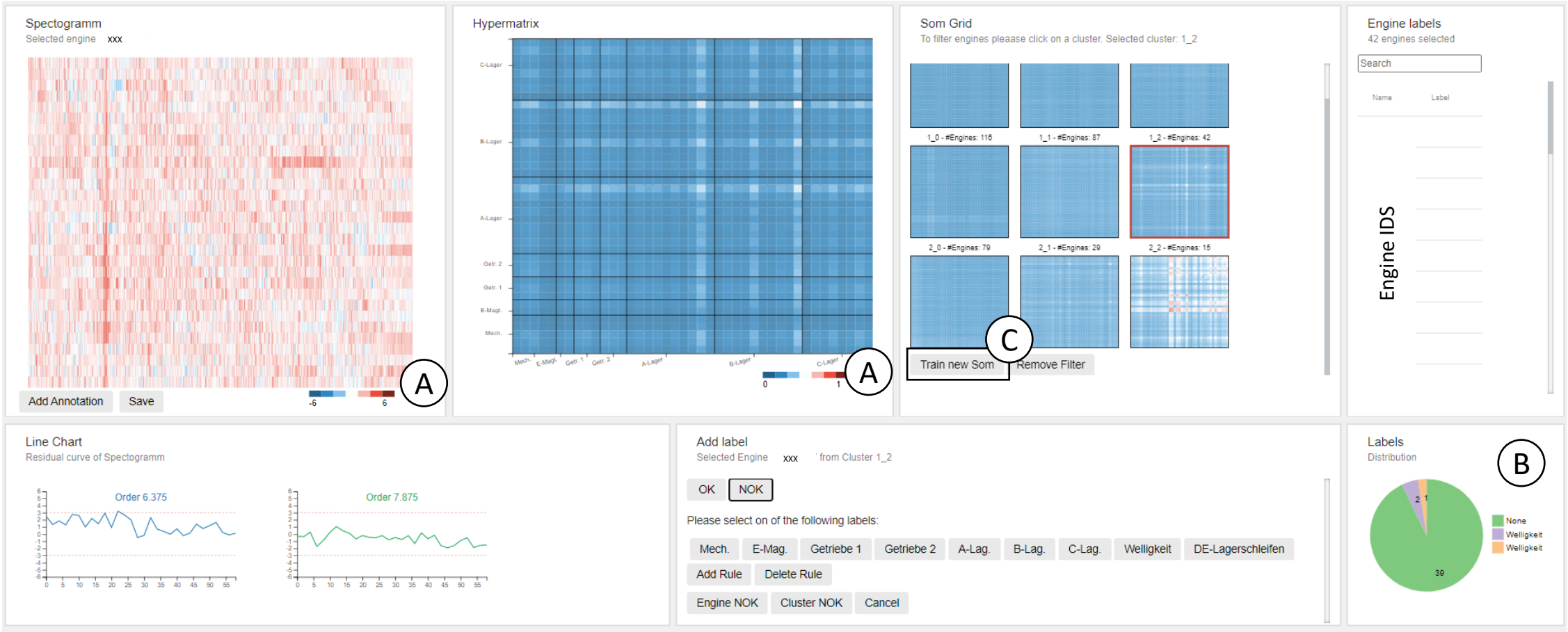
Iteration 2



Since, showing only submatrices of single engine components in the SOM grid resulted in a significant information loss, we showed the entire aggregated mean Hypermatrix for each SOM grid (A1). We also implemented an initial annotation functionality (B), where only one annotation at a time was possible. In this iteration, we also created a completely new SQL database and according API to better store and reuse externalized knowledge.

Color Theme and Label Distribution

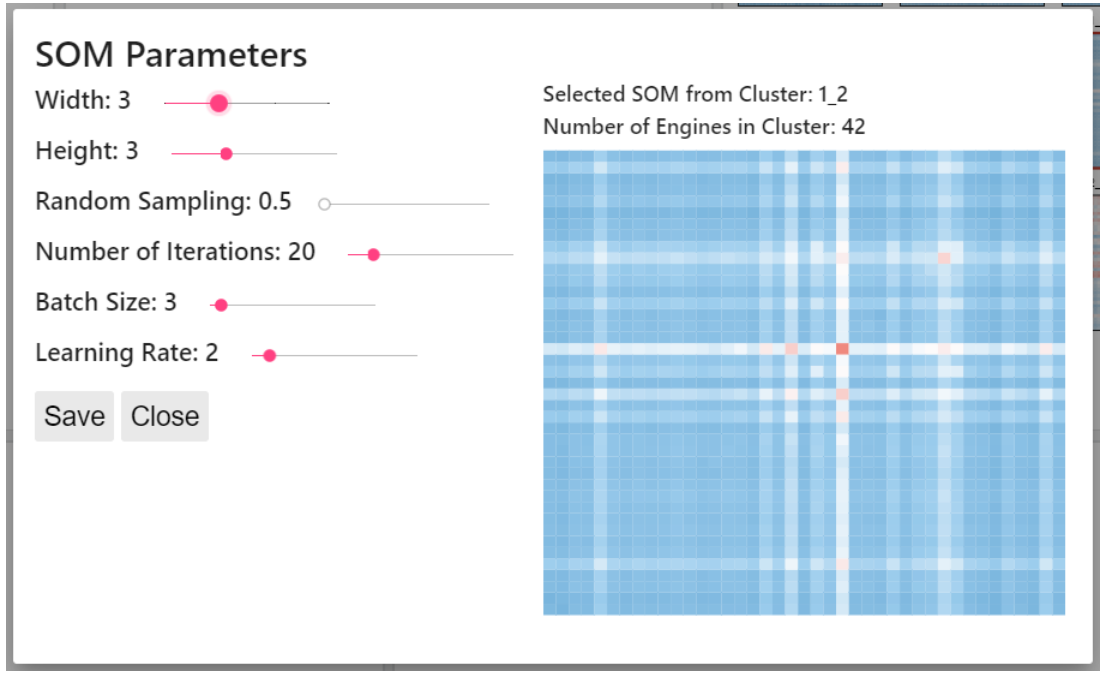
Iteration 2



In this version, we experimented with initial color themes (A). The presented color theme resulted in not very well distinguishable outliers in the Hypermatrix and the spectrogram. We also added a pie chart (B), to show the distribution of already entered labels. Also, an initial retraining of the SOM was implemented (C).

Initial SOM Training Dialog

Iteration 2



The first SOM training dialog shows only one aggregated hypermatrix for the selected cluster from the main view. However, parameters can already be set individually.

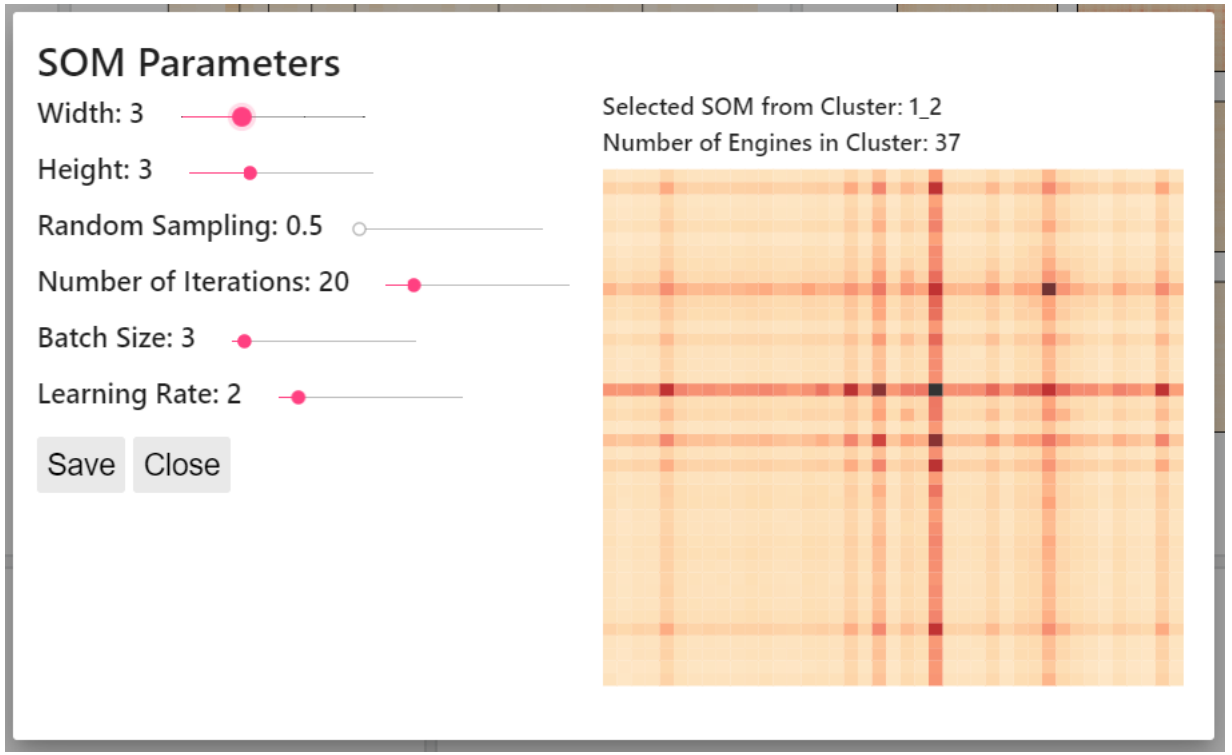
Visualizing Annotations of Previous Analyses

Iteration 3

Here, we improved the color scale both for the Hypermatrix and spectrogram (A). Now anomalies are better detectable in both views. We also visualized previous annotations from our lead users analyses (See gray rectangles in spectrogram (B)). Hence, domain knowledge was externalized and made available for other users.

SOM Training Dialog With New Colors

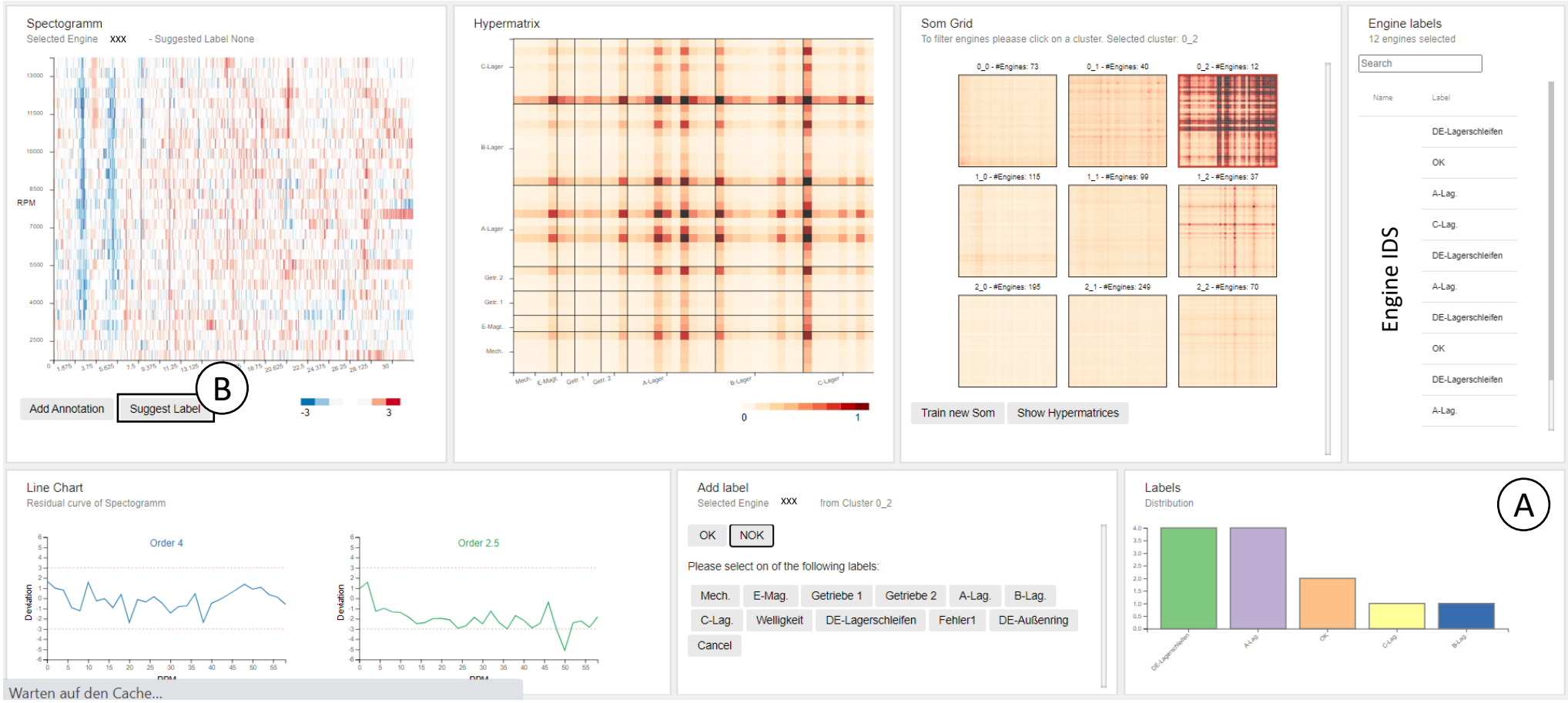
Iteration 3



Adapted SOM training dialog with new color theme.

Better Visualization for Label Distribution

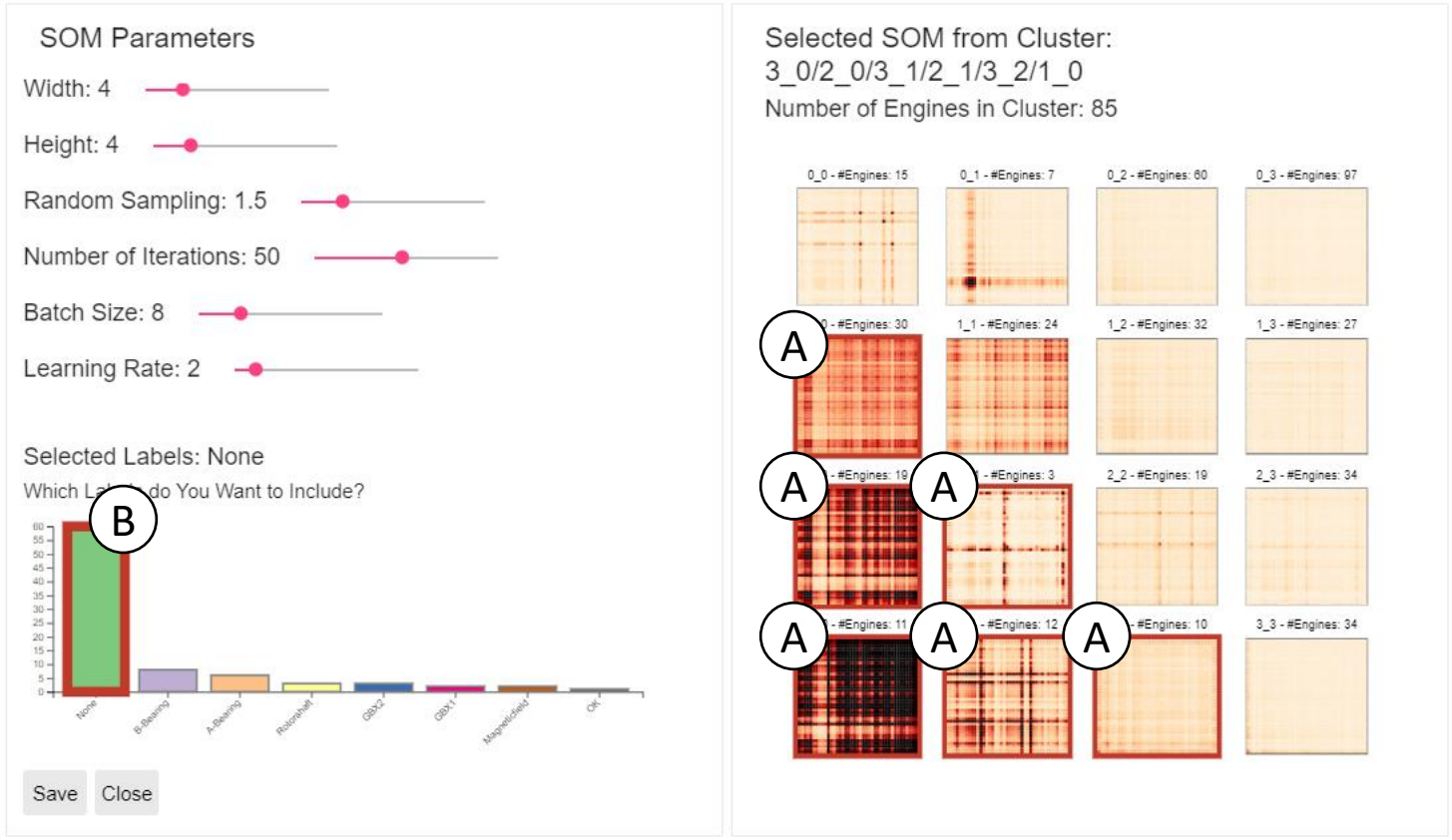
Iteration 3



Since pie charts are not appropriate if more than six categories are visualized, we changed the pie chart to a bar chart (A). Also more than one annotation per engine was possible and the label suggestion algorithm was implemented (B).

SOM Training Multi Cluster and Label Select

Iteration 3



Here, we included multi select for different clusters (A) and labels (B).

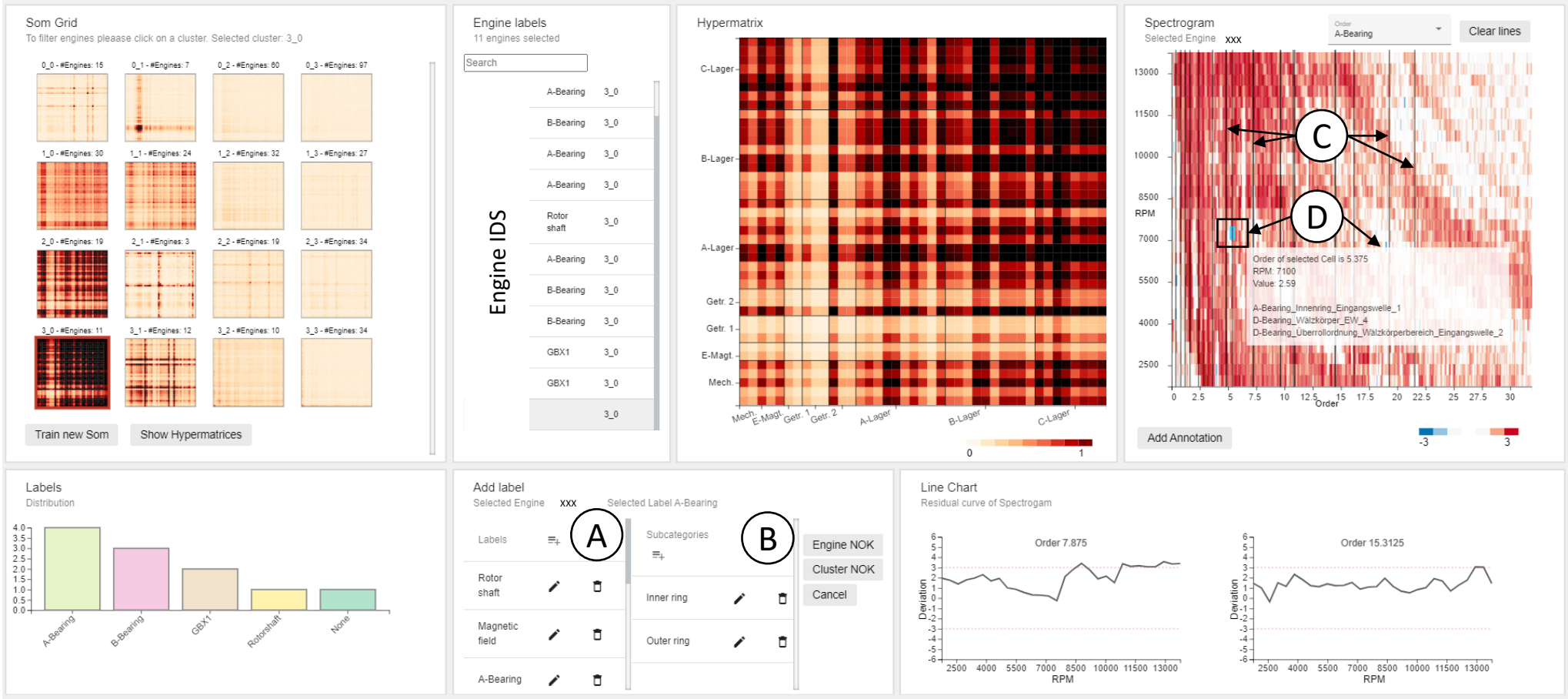
Closer Coupling of Hypermatrix and Spectrogram

Iteration 3

To better couple the Hypermatrix with the spectrogram view, now order lines from a selection in the Hypermatrix are shown (A). We also worked on our color theme for IRVINE, using colors only where necessary and removed in the line charts (B). In a later stage, we also removed the colors from the label bar chart (C).

Implementing User Feedback

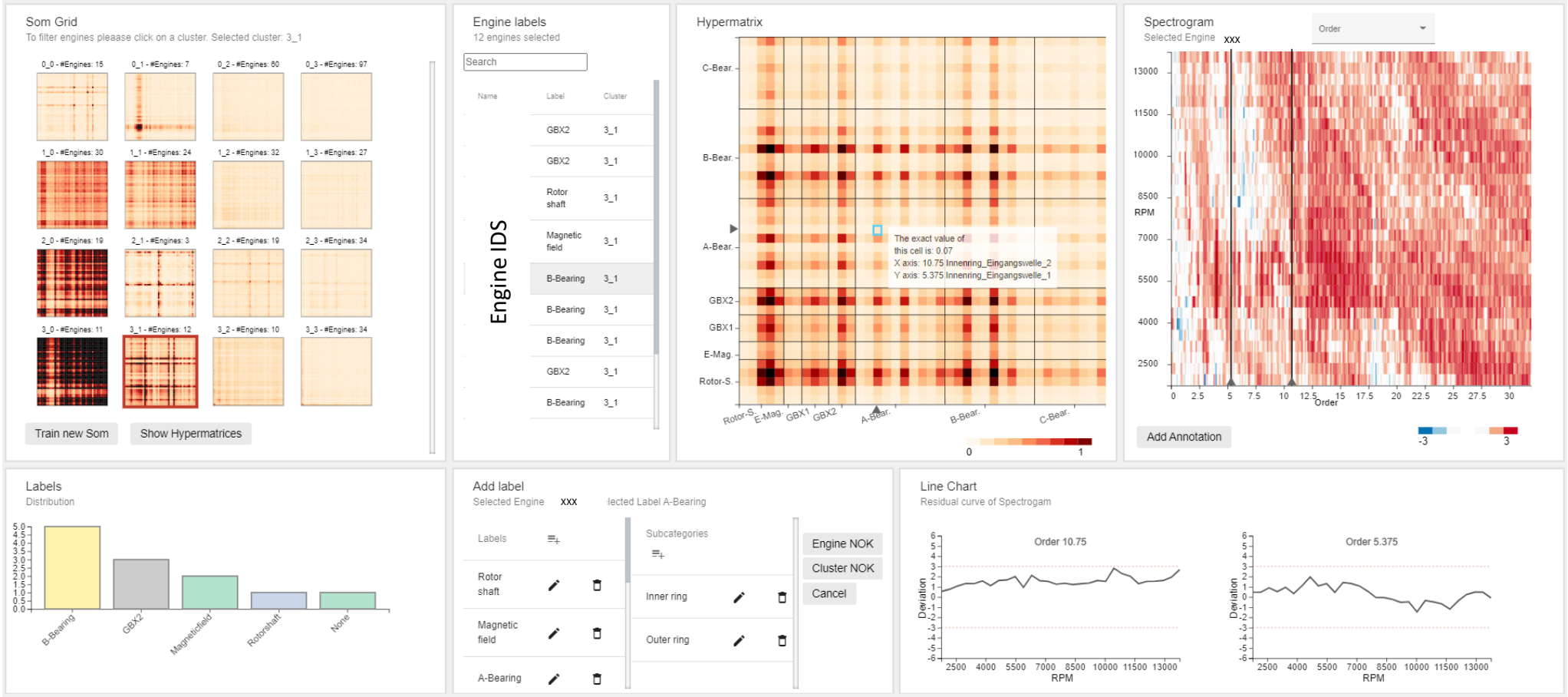
Iteration 4



After the evaluation with four domain experts, several features were suggested and implemented. First, the order of views was adopted to a smoother workflow from upper left to lower right. Labeling was changed from buttons to a table view (A) and label subcategories were introduced (B). Also metadata, such as all orderliness (C) for components (e.g. A-bearing) and detailed information about the region of the hovering event in the spectrogram (D) can now be shown in the spectrogram via black lines and the tooltip.

Changed Rendering in Backend

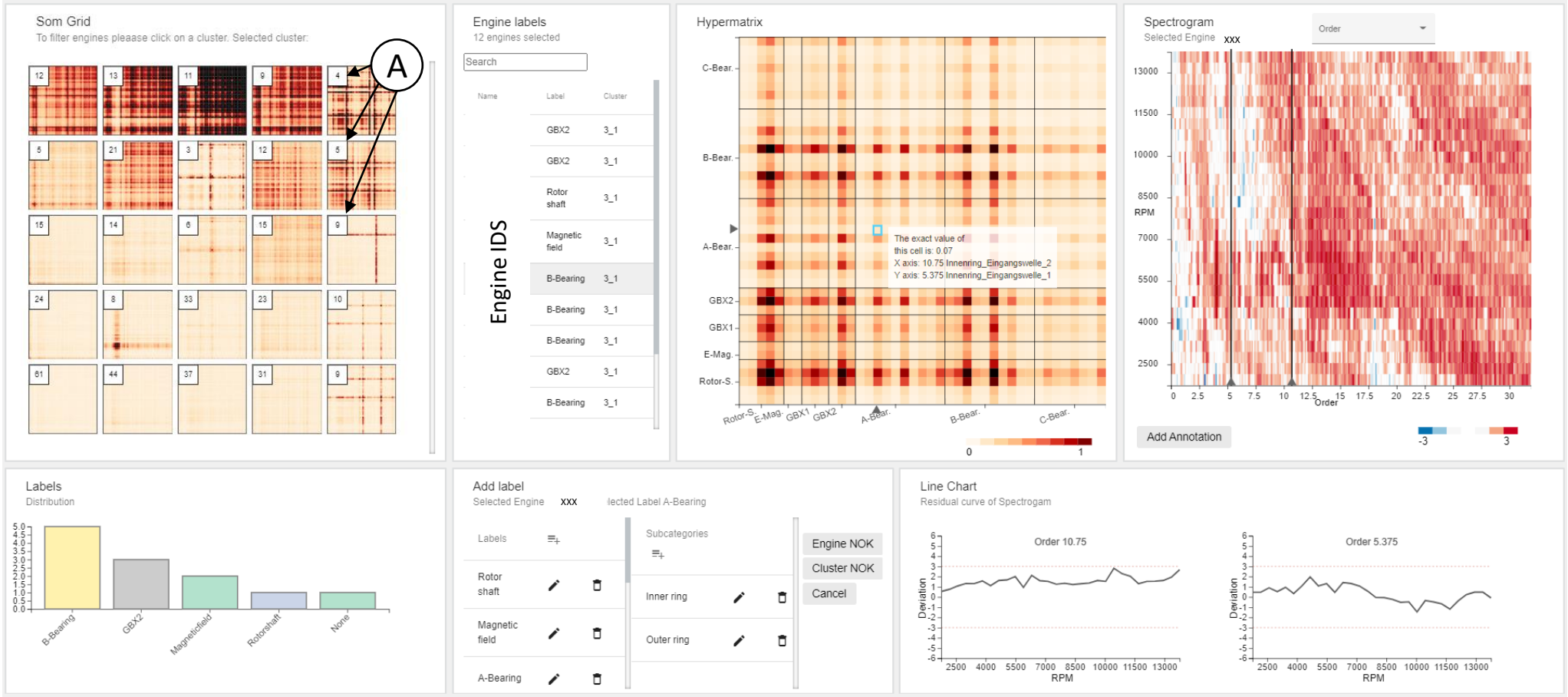
Iteration 4



This might not look as much, but here we changed our entire backend from d3 svg rendering to canvas rendering. Canvas rendering shows every object as bitmap making the entire rendering much more efficient. In our case, rendering speed was cut down from more than ten seconds for showing all views for one engine to less than a second. This was a major issue experts from our evaluation requested to solve.

Working on Small Multiples for Clusters

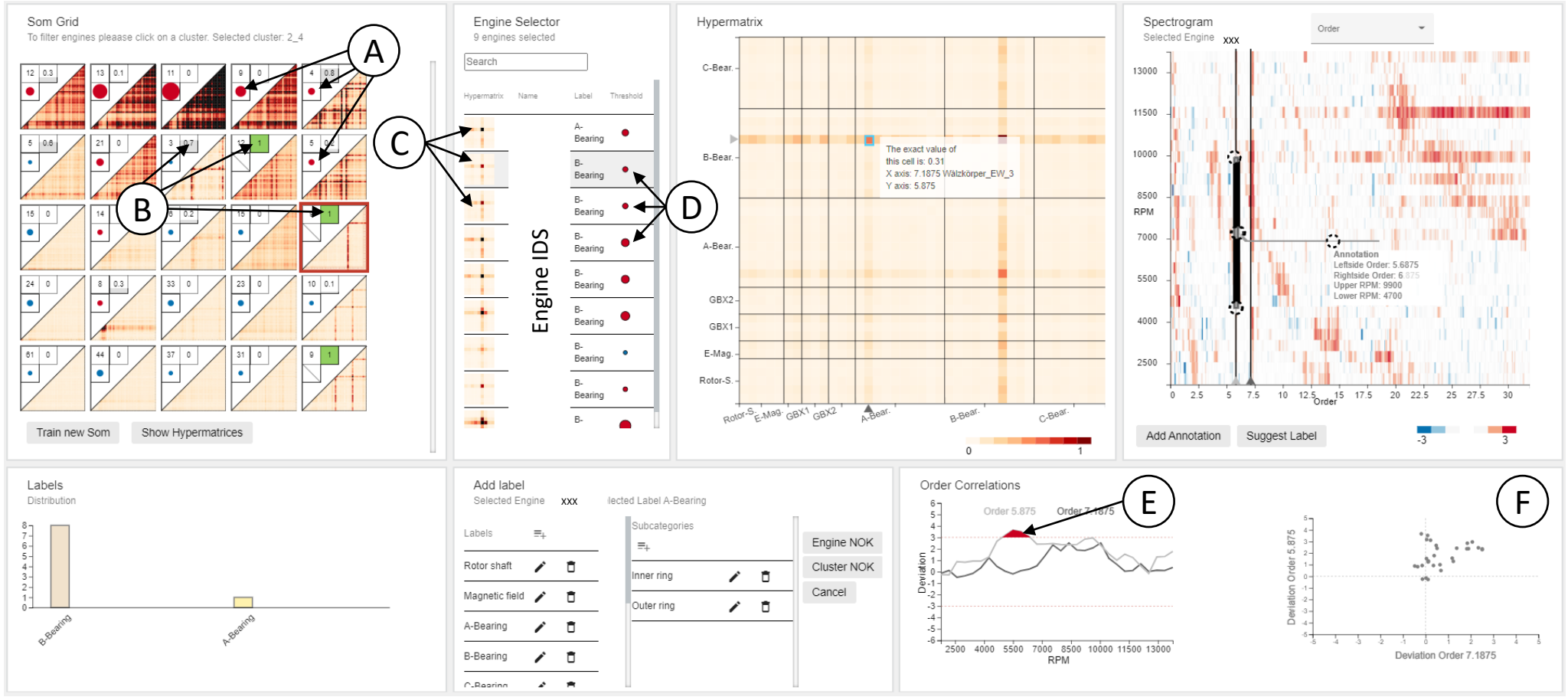
Iteration 4



After discussions with other visualization experts, we started working on including small multiples (A) to get a better overview over clusters and engines. Here, we included small rectangles showing the number of engines in each cluster.

Still Working on Small Multiples for Clusters

Iteration 4



We then included a glyph representation of aggregated anomalies in each cluster (A) and percentage of already labeled engines (B). We also included an aggregated hypermatrix for each engine in the table view (C) and aggregated anomaly score (D) for each engine. Furthermore, we colored threshold violations red (E) for all lines above three and blue for all below three in the line chart and included a scatterplot (F) to facilitate the analysis of correlating order pairs.

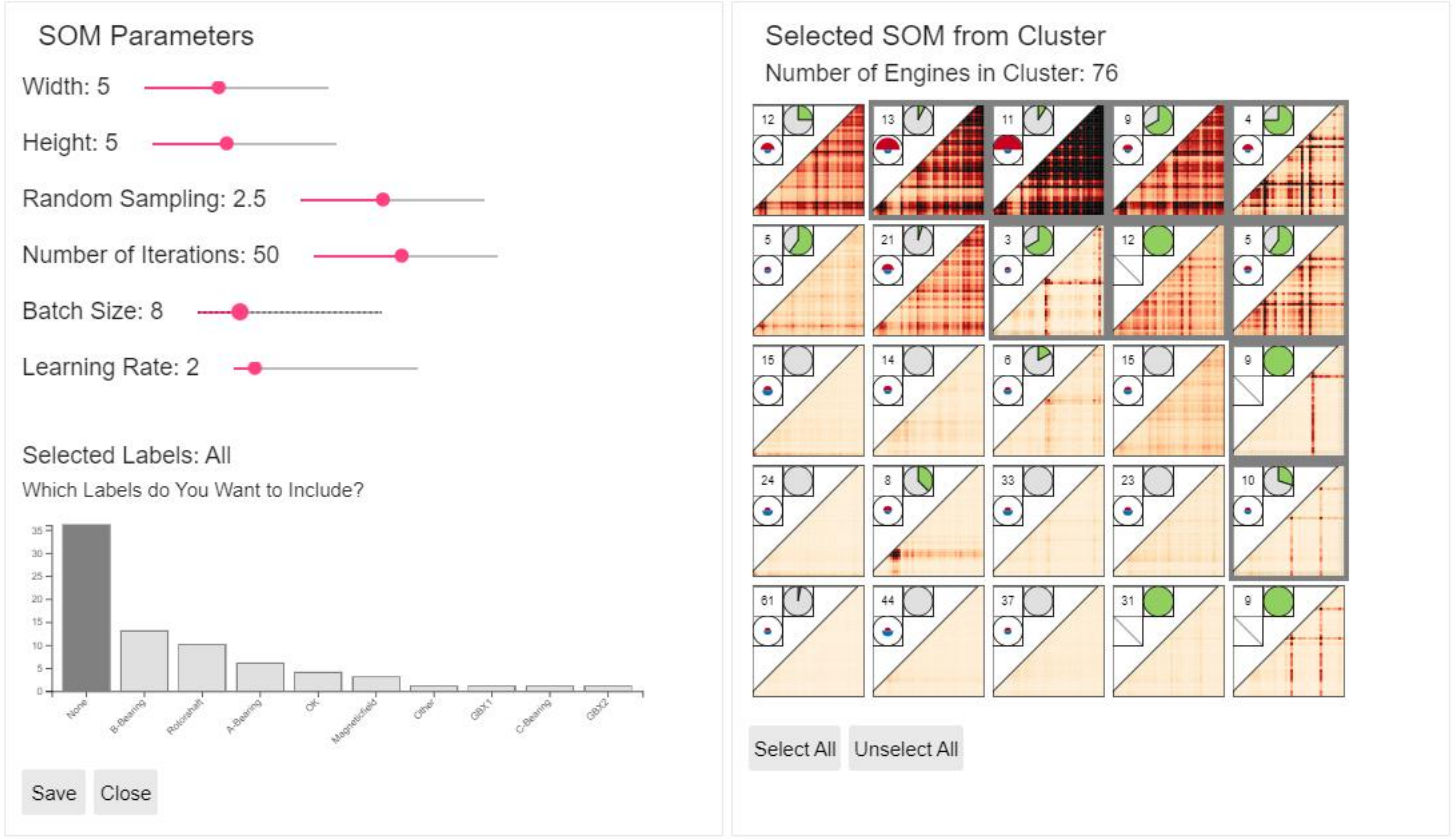
Finished Small Multiples

Iteration 4

A pie chart representation of already labeled components in the SOM grid proved to show more clearly how many labels are actually provided for each cluster (A). Since deviations below zero were not accounted before we included a glyph representation to show anomalies above zero in red (half red circle in SOM grid and table view (B)) and below zero in blue (half blue circle). We also removed colors from the label bar chart (C), because our users reported them to be confusing.

Final SOM View

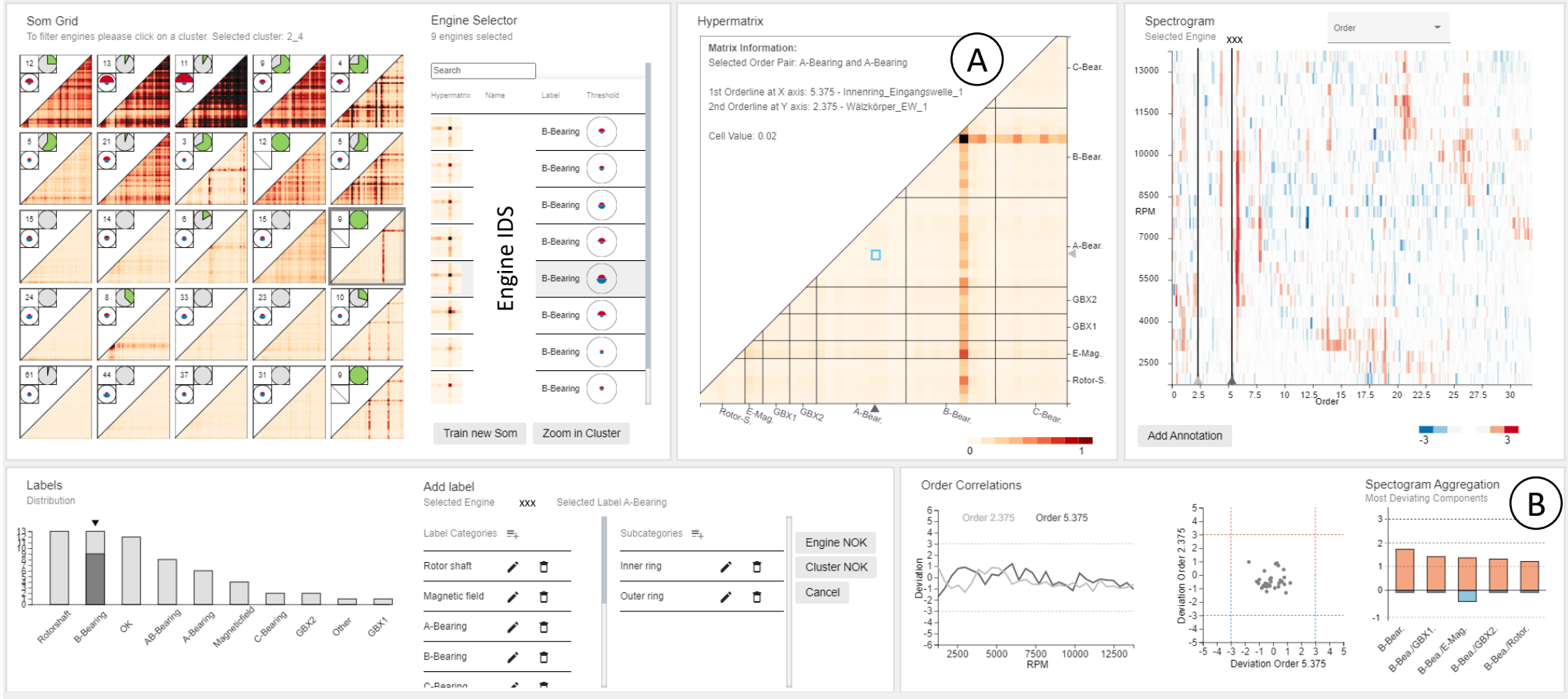
Iteration 4



For the final SOM training view, we included a “Select All” and “Unselect All” button and adapted the visualization according to the main view (Small multiples for clusters and greyscale for labels).

Part Aggregation

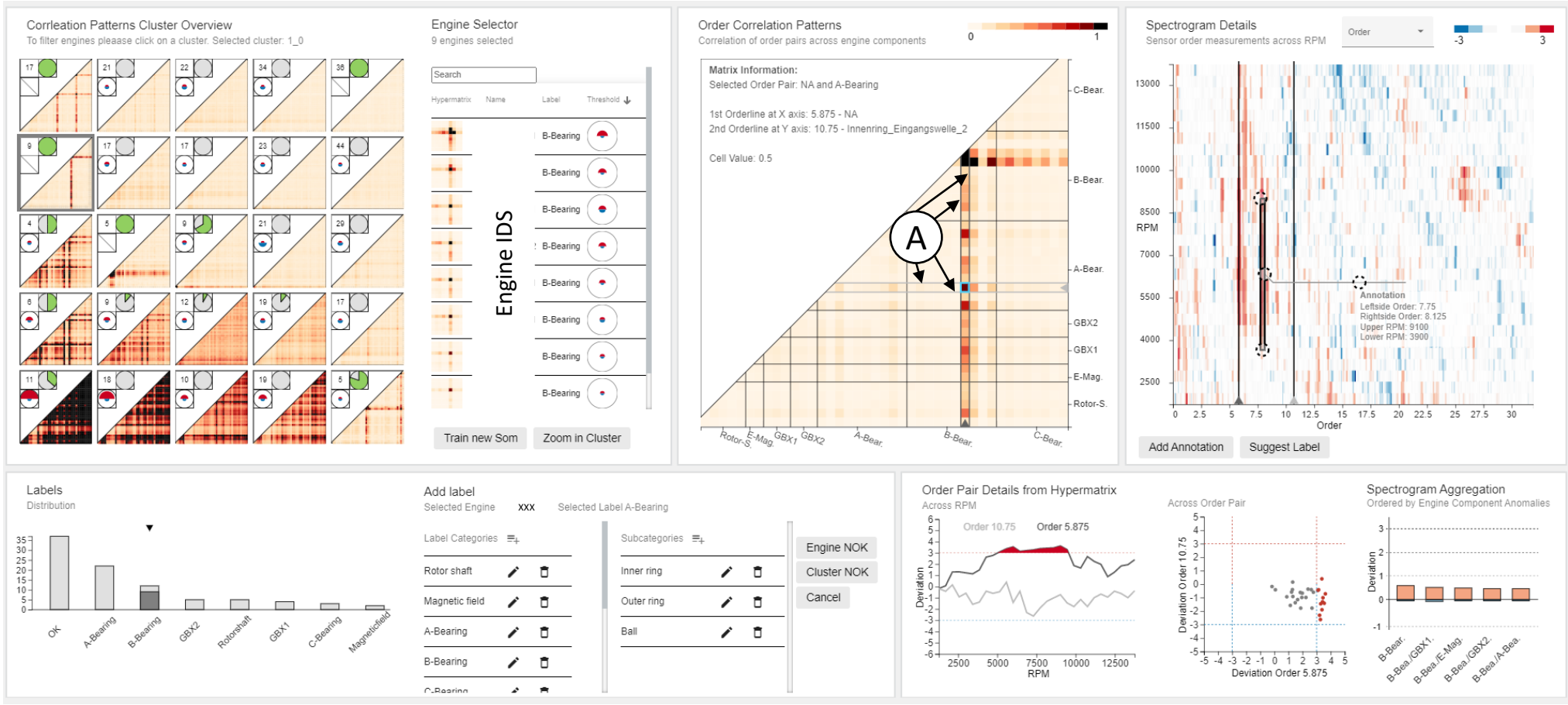
Iteration 4



Since Hypermatrices are symmetric, we can use only half of them with no information loss. We thus used the now available whitespace (A) to provide additional metainformation for the hovering event from the Hypermatrix. We also in included the five most abnormal engine components of each engine to better allocate the source of an error (B). The system you see here was presented to additional two domain expert to evaluate labeling speed.

Final System Deployed at BMW

Iteration 4



In the final, version we made some style and header changes and included lines to better see the hovering event in the Hypermatrix (A). VOILA we proudly present **IRVINE**!