Appendices

Appendix A. Standout Solution Details

This appendix gives more details on the leaderboard contributions. In the section header, we give the paper title along with their achieved ranks in the core/extended competition in parentheses. For each submission, we also give an inference diagram which in our opinion is supposed to summarize the approach from an information-flow perspective by linking to the trained models, the data used in training rather than to the architecture details and to the ensembling.

A.1. oahciy: U-Net + Multi-Task Learning

(Lu, 2021) presents an amazingly simple multi-task learning framework by randomly sampling data from all available cities and training the models to jointly predict future traffic for different cities. (Lu, 2021) uses ensembles by averaging 9/7 different U-Net models with varying architecture and seeds, all trained on all data for 4 training cities and 4 core cities. The models in the core competition are trained for 5 epochs, the models for the extended competition for 50'000 steps only.

(Lu, 2021) argues that the multi-task learning can be regarded as an implicit data augmentation and regularization technique preventing overfitting when trained on one city only and forcing to learn city-agnostic representations and improving data efficiency. (Lu, 2021) reports the results of a comparison with a series of domain-adaptation techniques and find that their approach is superior in both competitions and report to have tried different ensembling techniques. For temporal domain adapation, he conducted experiments by using 3 of the training cities as training data and 1 city (Bangkok) for evaluation; his findings suggest that training on the target city is crucial in comparison to more data from the 3 training cities and that adding 2019 and 2020 data for at least one city is crucial encouraging the model to learn to adapt to temporal domain shifts during training. With respect to the core competition, they achieved good performance in the extended competition by reducing the number of parameters in the U-Net model and adopting an early stopping strategy. (Lu, 2021) suggests performance could be improved by fine-tuning models in a city-dependent manner and using manually designed features like time of day.

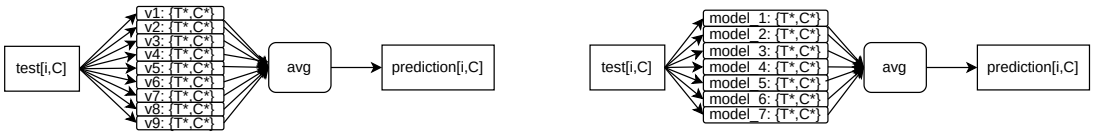


Figure 10: Inference oahciy (Lu, 2021) (left: core competition, right: extended competition).

A.2. sungbin: U-Net Ensemble

The approach of (Choi, 2021) is very similar to (Lu, 2021), also using different U-Net architectures and averaging ensembles. In contrast to (Lu, 2021), (Choi, 2021) trains on target city-dependent training data, too, resulting in ensembles of 16/4 models in the two

competitions; each prediction comes from an ensemble of 7/4 models, of which 3/- are city-dependent. (Choi, 2021) reports that in previous participations (Choi, 2019, 2020) having training data from different cities hurt performance; (Choi, 2021) suggests that the additional cities for training only make the difference; however, no ablation studies are presented for this claim. As Berlin and Istanbul are in the core competition target cities as in the previous competition, we miss the exploration of the performance of last year's approach and of the contribution of static road data.

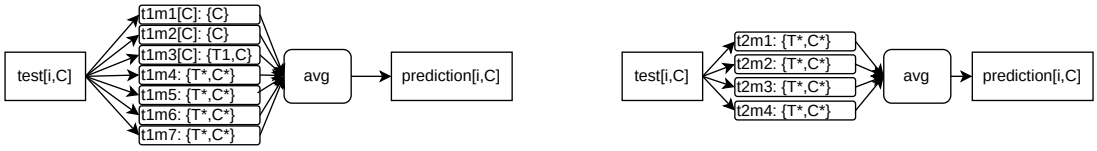


Figure 11: Inference sungbin (Choi, 2021) (left: core competition, right: extended competition).

A.3. sevakon: U-Net with Temporal Domain Adaptation

(Konyakhin et al., 2021) also base their approach on the success of U-Nets in previous competitions. However, in contrast to (Lu, 2021) and (Choi, 2021), they train their models on the target city in the core competition only (they did not participate in the extended competition). They use three different architectures (vanilla U-Net, DenseNet, and EfficientNet pre-trained on Imagenet (Deng et al., 2009)), a static mask derived from dynamic data and a per-pixel and per-channel temporal domain-adaptation factor. Their final prediction is derived from the 3 models; each model is used with and without TDA, resulting in 6 predictions to which the static mask is applied and which are then averaged. They find slightly better results are achieved by using a static pixel-wise binary mask generated from the training and test data for non-zero values among all channels. In addition to their TDA heuristics, (Konyakhin et al., 2021) also investigated pseudo-labelling without improvement; pseudo-labels are generated by applying the model on the (unlabelled) samples from the target domain and use these in training; pseudo-labelling had been used as an entropy regularization for probabilistic multi-class classification tasks (Jaiswal et al., 2019; Lee, 2013).

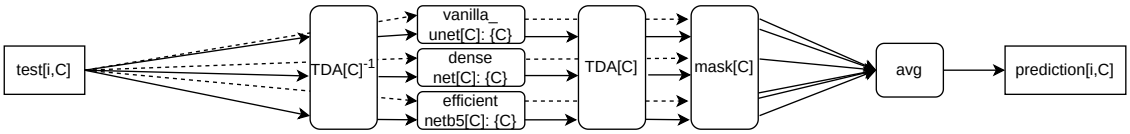


Figure 12: Inference sevakon (Konyakhin et al., 2021) (core competition). The city mask is derived from the training and test data.

A.4. nina: U-Net++ on Patches

(Wiedemann and Raubal, 2021) also use a U-Net variant, but in patch-based manner, as it was shown beneficial in other segmentation tasks (Zhang et al., 2006; Ghimire et al., 2020)

(also (Misra et al., 2020) in classification), without using the static road information. Using patches allowed them to use parameter-heavier UNet++ with more skip connections (Zhou et al., 2019, 2018), which they suggest to have been helpful in light of the sparsity of the data. They subsample from the available labelled data, processing 1000 patches for two epochs and then re-sampling (10 patches from 100 files). At inference time, the predictions are built by averaging the per-cell predictions from all patches that contain the cell.

They compare different model variants and different patch sizes and strides; their best performance was achieved in both competitions with quadratic 100×100 patches and stride 10. They show that the ensemble-like behaviour of patch-wise prediction accounts for small but significant increase in performance with smaller strides; they argue that the main advantage of the method lies in the simplification of the problem by splitting the data into smaller parts. Furthermore, (Wiedemann and Raubal, 2021) did an error analysis: the overwhelming part of the error is due to the predictions in the speed channels and volumes are zero most of the time. They further looked into those cells with non-zero volume data and derive speed MSE on those cells, which seems to be "hardly better than chance", at least for a small validation set from the labelled data. They even investigated into training with a loss function that masked out speed on zero ground truth volumes. At test time, the speed of zero-predicted-volumes is imputed with zero. However, the current competition metric does not reflect the joint volume-speed distribution directly and under MSE different speed predictions might be less penalized. In discussion, the authors also pointed out that MSE masked this way might incentivize models to learn to predict "free flow speed" (*i. e.* speeds taken without the presence of other vehicles, see *e. g.* https://en.wikipedia.org/wiki/Fundamental_diagram_of_traffic_flow) in case of zero volume or a speed appropriate to the current traffic situation and not needing to gamble for the data collection gaps (as the GPS probes come from vehicle fleets representing only a part of total traffic).

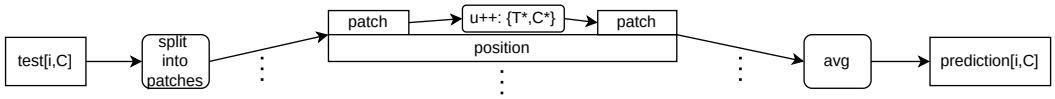


Figure 13: Inference nina (Wiedemann and Raubal, 2021) (core competition and extended competition).

A.5. ai4ex: SWIN-Transformer

(Bojesomo et al., 2021) uses a Swin-UNet structure where all convolution blocks are replaced by shifted window self attention; downsampling in the encoder is achieved by trainable patch merging layers and upsampling by patch expanding layers in the decoder branch; skip connections are implemented by a combination of addition and concatenation. The paper compares 3 different configurations with different embedding dimensions, feature mixing to find the best model in the leaderboard.

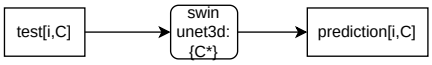


Figure 14: Inference ai4ex (Bojesomo et al., 2021) (core and extended competition).

A.6. dninja: Graph-Based U-Net

(Hermes et al., 2022) are the only graph-based contribution in the competition; they aim to leverage prior knowledge on the underlying structure of the street network, ignoring areas without any traffic information, and thereby to achieve better generalization and transfer by using a graph-based approach. Vanilla graph layers do not capture 2-dimensional topology like CNNs do – in order to capture this information, they enhance existing graph layers by using 4 subgraphs corresponding to the 4 headings of the challenge data. They show that these subgraphs are consistently beneficial for all cities; edge features contain a CNN-based embedding of the road graph as image; a global state vector contains the summed-up and scaled node features and encoding of time of day and day of week. (Hermes et al., 2022) use these Graph layers in a UNet architecture, where the 2D node positions are use for up and downsampling on graphs: downsampling works by taking the feature-wise max of node and edge features, while upsampling works by introducing edges from the input graph to the target graph and using graph propagation. This up- and downsampling approach effectively expands the receptive field of the graph approach without requiring too deep GNNs, which are thought to lead performance drop at a certain depth. Although ablation studies and extensive hyperparameter tuning are missing in their work, the work seems promising.

A.7. resuly: 3DResNet and Sparse-UNet

(Wang et al., 2021a) use 3DResnet (Wang et al., 2021b) with 3D convolutions in 4 residual block and an output block of sequential CNN layers to restrain the temporal relationship in the core competition and Sparse U-Net (Graham, 2014; Choy et al., 2019) with data in Coordinate Format (COO) for the extended competition. They enhance data loading by two-level shuffling over day files and then over indices, randomly picking up a number of files as training samples in each epoch. To ensure spatial and temporal diversity, these day files need to include all available cities in 2019 and 2020 and cover all seven days of the week. They show large training speed-ups for the Sparse UNets. Comparing the two approaches by training on each city separately, the find no consistently better method, with considerable differences in the convergence behaviour and the final loss achieved; above all, they conjecture that sparse UNets generalize better on the extended challenge but perform worse on the core challenge, however without giving systematic comparison of the same method applied in both settings.

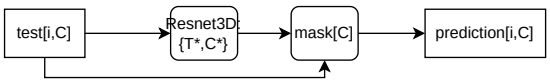


Figure 15: Inference resuly (Wang et al., 2021a) (left: core competition, right: extended competition). The mask is derived from the test data.

A.8. jaysantokhi: Dual-Encoding U-Net

(Santokhi et al., 2021) use a dual encoding U-Net architecture aiming at a lightweight approach for real world deployments containing significantly fewer parameters and shorter training times. The architecture consists of two encoders one of which has skip connections

to the decoder; encoder and decoder consist of Convolutional LSTM layers. The skip connections are not vanilla, but designed to carry the hidden and cell states of the encoder LSTM to the decoder LSTM, which is crucial for the approach. In both competitions, 4 models are pre-trained on the training cities and fine-tuned on the core competition cities. In the core competition, the city-specific fine-tuned model is used, whereas in the extended competition an architecture with fewer parameters is used and predictions are built by averaging over the outputs of all 4 models; in addition, in the core competition, the model outputs directly 6 frames, whereas the model in the extended competition outputs 12 frames. In both competition, a mask derived from the test data was found to be more performant than the static mask or mask based on the training data.

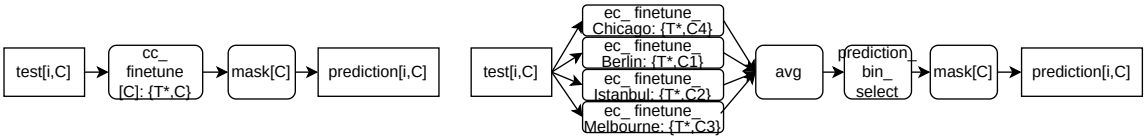


Figure 16: Inference jaysantokhi (Santokhi et al., 2021) (left: core competition, right: extended competition). The city mask is derived from test data.

Appendix B. Relating MSE to the Variance in the Data: Where do Models Struggle?

Here, we detail on Section 3.2.

For the analysis, we took standard deviation on the ground truth data of the 100 tests of the core competition (*i. e.* 18 time bins each, the 1 hour input and the 6 prediction slots from the 1 hour prediction horizon, see (Eichenberger et al., 2022) for the test slot sampling details). We compute MSE for each directional pixel ($863280 = 495 \cdot 436 \cdot 4$ in total), separately for volume and speed. Finally, we plot MSE per std bin to find "hot spots".

B.1. Relating speed variance to speed MSE

We have a strong asymmetry in the MSE levels in both competitions (see Figure 32), *e. g.* for Lu (2021), the winner of our core competition, we have speed_mse: 148.427, vol_mse: 10.440, all_mse: 79.434 for MSE on the city of Berlin only (plots and data can be found in Eichenberger and Neun (2021)). Therefore, we put the focus of our analysis on relating speed variance to speed MSE. If we bin the test data by speed standard deviation as in Figure 17, we see that most directional pixels have little standard deviation. This is indicated by the gray line going down almost monotonically showing the counts for every speed std bin. Also notice that we do not show the full y range on the left.

If we relate this to MSE for speed of the submissions in the core competition, we see that speed MSE per directional pixel in the bin is going up almost monotonically and faster than in a linear fashion Figure 17. The curves of the different participants are pretty in parallel, so it looks as if they struggle at the same places. The different participants are represented by different colors.

If we multiply the two curves of Figure 17 top, we get the dashed curves of Figure 17 bottom, which shows the sum of the MSEs of all directional pixels in the bin. This allows us to see which speed std areas contribute most to the total speed MSE. The monotonic growing solid curve are the cumulative sums of the dashed ones, containing the same information as the dashed ones. A peak in the dashed curves is reflected by steepness in the solid curve. The final value top right is the sum over all $(495, 436, 4)$ virtual speed detectors of the city, summing the per-pixel and per-heading MSE over all 100 test slots. If we divide the final cumulated number by the the number of such directional pixels, we get the speed_mse of 148.

We find two critical ranges of speed std where most of the final MSE is accumulated: around 35–55 and around 105–125. In the lower critical range (35–55), average MSE per speed std bin more than, doubles and increases monotonically, while the number of pixels per bin is slightly decreasing, which cumulates heavily; so despite the medium average MSE level, this translates into a peak in the per-bin cumulated MSE bin curves. The higher band (105–125) covers very high average MSE, but at a low number of pixels, which also translates into a peak.

B.1.1. Mapping out the two critical ranges

We now visualize the locations of these two critical speed std ranges.

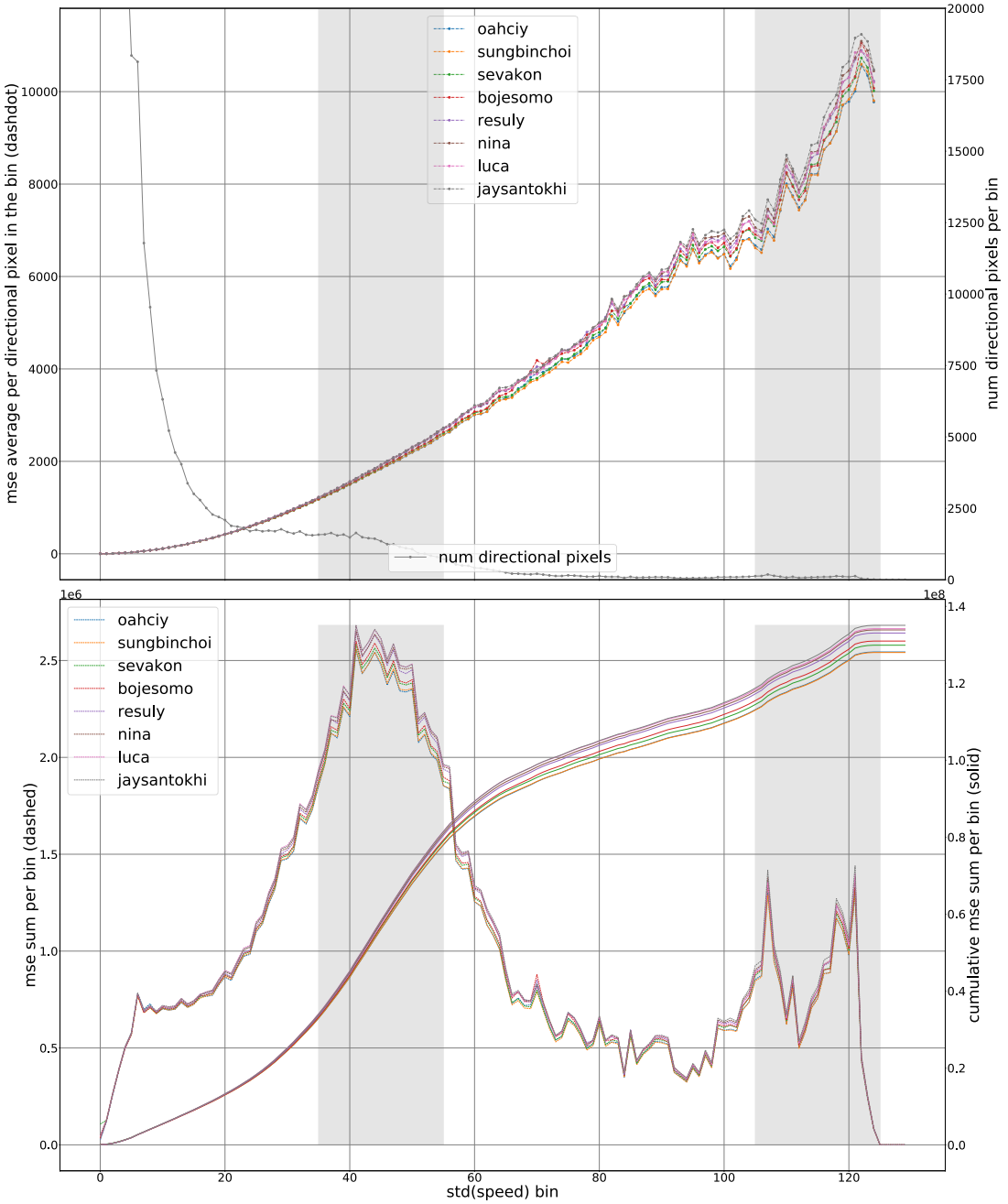


Figure 17: Relating MSE to std for speeds in BERLIN core: distribution of std among directional pixels (axis capped) and average MSE (top); summed MSE and cumulated summed MSE (bottom).

Figure 18 shows the locations of the 35–55 speed std band on the left side. This covers the main arteria as well as parts of long-range country roads where commuting traffic starts early in the morning. So these may be areas that are naturally hard to predict. This stems on the one hand from medium "free flow speeds" and high speed variance or no data, and on the other hand from relatively high "free flow speeds" and a lot of no-data in off-peak times.

The 105–125 speed std band on the right side in Figure 18 shows areas with usually high speeds and some no-data during the night.

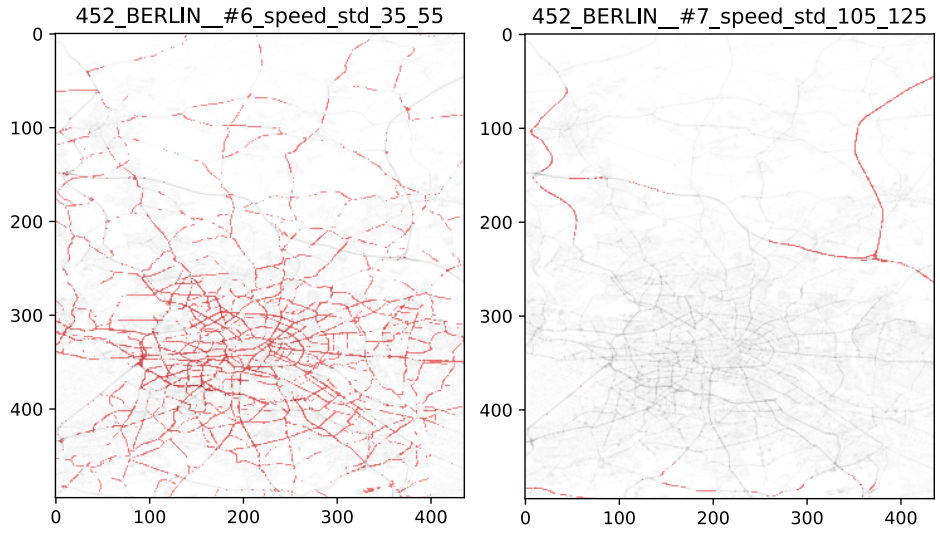


Figure 18: Locations (red) of the 35–55 (left) and 105–125 (right) speed std bands

B.1.2. Revisiting 3 Berlin locations

If we take each cell and heading as a speed detector, we have $495 \cdot 436 \cdot 4$ virtual speed detectors. We can now plot the mean against the std speed for data collected from the test slots of the core competition, see Figure 19. Each dot corresponds to one cell in the grid

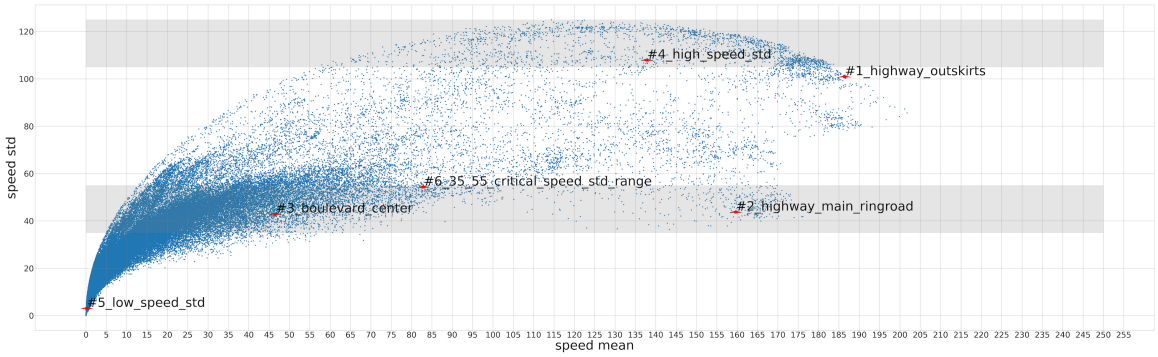


Figure 19: Critical ranges and sample situations in Berlin.

and one heading. Since a lot of cells never encounter any GPS probes, most of those points are close to the origin of the mean-std coordinates system. We choose 6 Berlin locations in Figure 20. In Figure 21, we plot volumes and speeds for one sample day for each of them and discuss these illustration with respect to the two critical ranges.

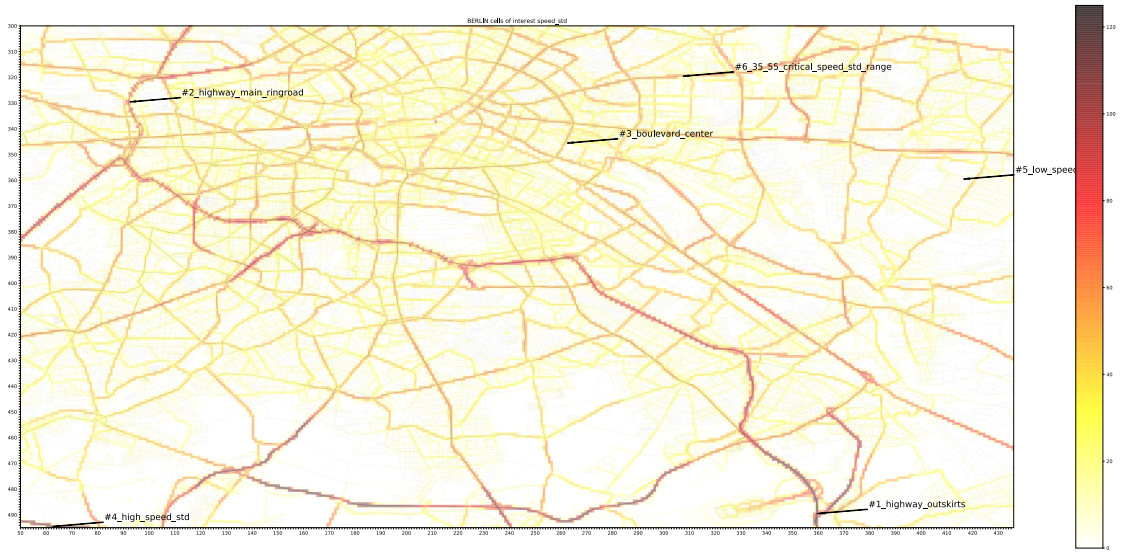


Figure 20: Speed std heatmap for Berlin showing the max of all 4 headings per pixel, showing the 3 Berlin locations.

#1 is a highway in the outskirts has high speed mean and high speed std. This reflects the high speeds during day time and the data sparsity during night time. #1 is close to the 105–125 critical range; it has many no-data points and the average speed is even higher, so every no-data is penalized a lot if the prediction is too high.

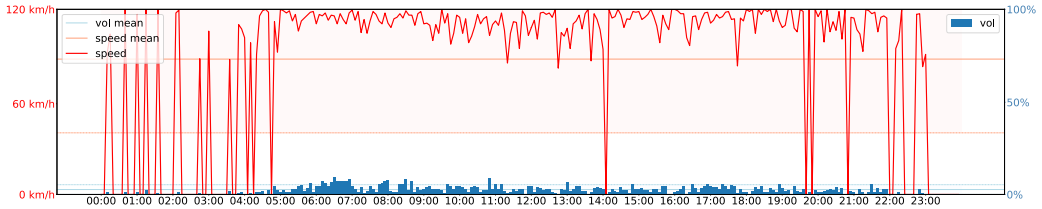
#2 is a main ringroad highway, has high speed mean and relatively high speed std. Although, there is more data in the night, the sporadic speed drops make standard deviation still pretty high. Pretty high free flow speed with a few or no-data points is hard to guess, as every no-data is penalized considerably if the prediction is too high. The zero-volumes during the first hours of the day might be a production artifact, but the location has sparse traffic during the night in all other days we checked.

#3 is a boulevard in the center of Berlin, which shows that moderate free flow speed and frequent no-data can make the prediction task hard, too.

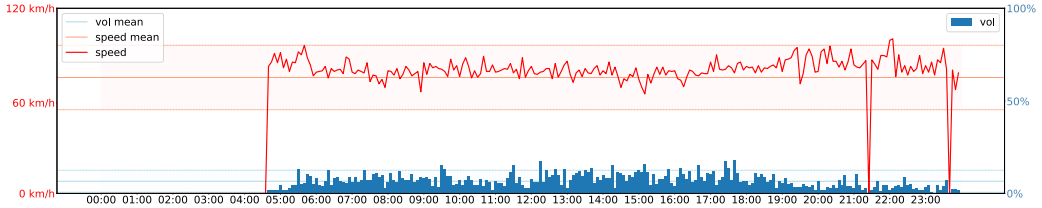
#4 is an example from 105–125 speed std critical range, which translates visually into a very spiky speed curve over day. Notice that volumes are low and the spikes mainly come from the no data points. The variance in the speed measurements masked on non-zero volume would be much lower.

#5 is an example of very low speed std. It shows that there are many directional pixels not or extremely sparsely covered with traffic data, either due to the fleet bias (not covering full traffic) or because there is no traffic in those regions at all.

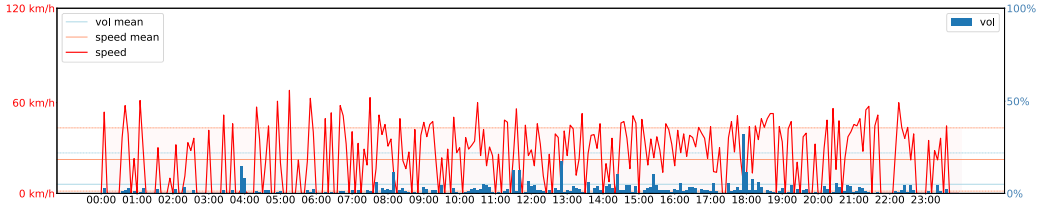
#6 is similar to #3, but with slightly higher volumes, lower speed levels and a clearer difference between day and night traffic, which is plausible for a location on the main arteria but not on a highway.



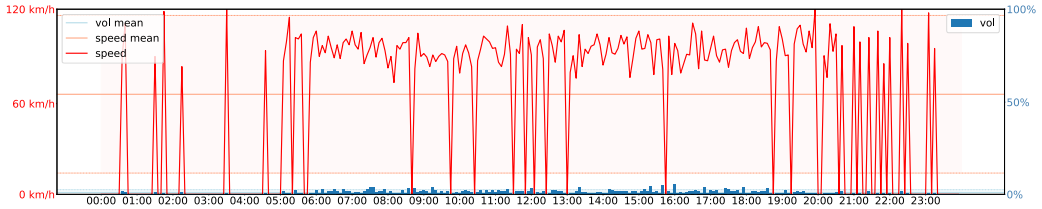
(a) #1 highway outskirts (489, 359, NE). vol: 6.78 ± 6.68, speed=186.46 ± 101.55



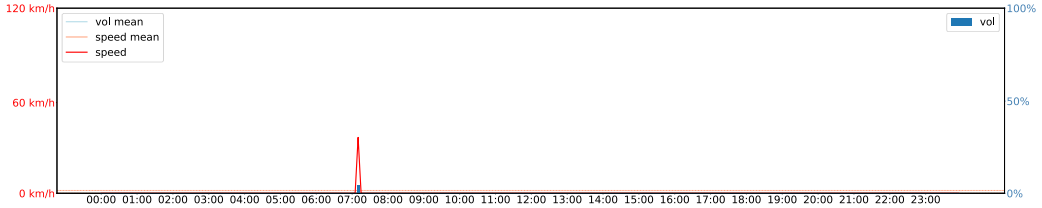
(b) #2 highway main ringr. (329, 92, NE). Vol: 16.80 ± 15.112, speed=159.69 ± 44.37



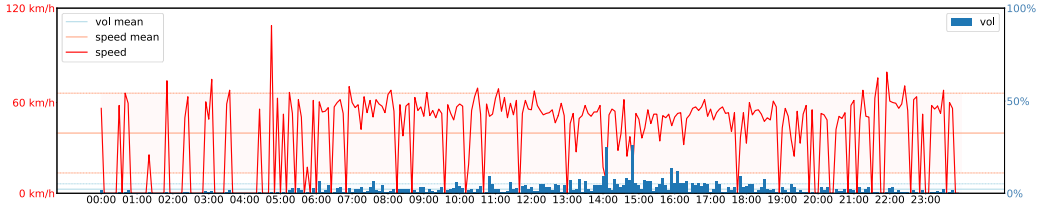
(c) #3 boulevard center (345, 262, NE). Vol: 2.57 ± 43.15, speed=46.69 ± 43.56



(d) #4 (494, 62, SE) in 105–125 speed std r. Vol: 2.78 ± 3.48, speed=137.87 ± 108.55



(e) #5 (359, 416, SW) with low std. Vol: 0.04 ± 0.73, speed=0.23 ± 3.69



(f) #6 (319, 307, NE) in 35–55 speed std r. Vol: 5.67 ± 7.35, speed=82.94 ± 54.94

Figure 21: Data from one day in the target domain of the core competition for the 3 Berlin locations and 3 sampled for low speed std, and the two critical speed std ranges. Speed curve in red and volume bars in blue, along with mean lines and std hulls, mean and std from all test slots of the core competition (input and ground truth).

B.2. Relating vol variance to vol MSE

The corresponding plot for volumes is Figure 22.

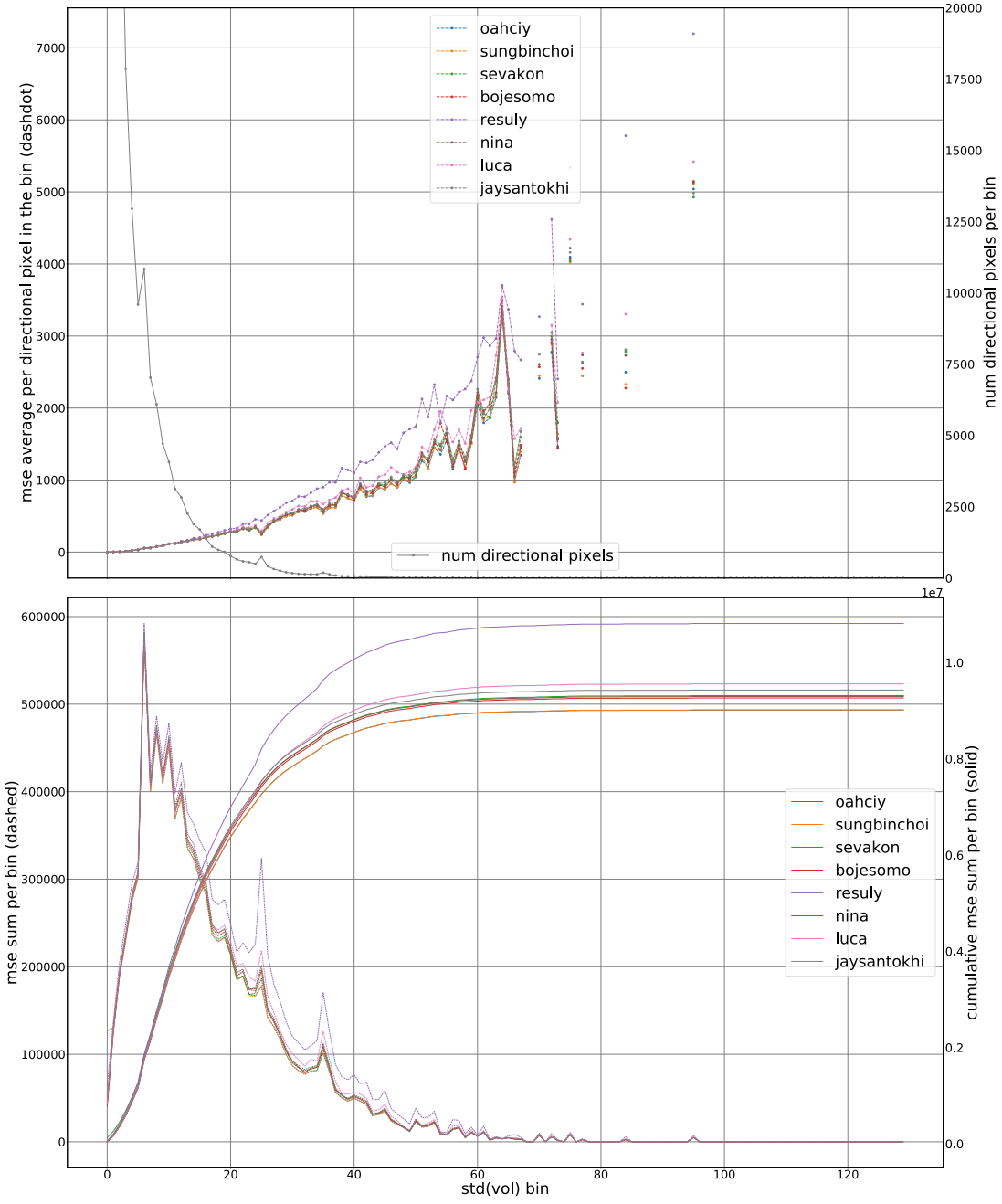


Figure 22: Relating MSE to std for volumes: distribution of std among oriented pixels and average MSE (top); summed MSE and cumulated summed MSE (bottom). The shaded gray areas highlights the two critical speed std ranges.

B.3. Limitations of the analysis and future work

As remarked in Section 3.2, MSE does not optimize each directed location independently, so the interpretation here has to be taken *cum grano salis*. Despite this limitation, we think it is relevant. Future work should address the following points:

- it is important to relate the binning of the directional pixels is based on std in the test slots (input hour and ground truth predictions), it should be checked whether this binning is robust when taking not only the sampled data for the test slots;
- conduct the analysis for other cities as well;
- in the analysis, the mean or percentiles in the data should be taken into account to put the interpretation of related the traffic patterns onto a more solid basis;
- in the same spirit, the analysis should be carried for day and night time separately and compared to the full day.
- in addition to looking at raw MSE values, it might also be interesting to calculate the per-pixel R^2 values for different models.

Appendix C. Outlier Special Prize

Here, we give more details on the Outlier Special Prize introduced in Section 3.4.

C.1. Outlier Special Prize

Quality of traffic prediction heavily relies on the performance in anomalous situations. We aim to have a first look at how models from the t4c21 perform in those situations, both quantitatively as well as qualitatively. Even without sparsity due to vehicle fleets nor daytime nor regional sparsity, outliers are rare in the data and hence MSE does not give a lot of weight to them. Also, typically there are many plausible future scenarios in an outlier situation with a fat tail of scenarios. We invited all Summit/Symposium Participants of *Traffic4cast* 2021 to re-run their models on a new test set for the *Traffic4cast* 2021 Outlier Special Prize. Every participant was allowed to submit two predictions: one with the "plain vanilla" model, as used for the original submission, and one with further training applied. The score was not disclosed to the participants and re-submission was not allowed. The participants were asked to make a prediction on the full without disclosing the outlier location nor the heading. As in the core and extended competitions, only time of day and day of week were disclosed. Also, we did not disclose whether the new slots would before or after the Covid shift.

C.2. Outlier Heuristics

We tackled outliers in a very pragmatic way. We were not interested in a general definition of outliers, but very much in finding some examples and plot what the models predict in these situations. The outcome of a few trial-and-error iterations was a heuristic that finds outlier situations in a single directional cell.

We compute quantiles on the data for each pixel and each channel separately and use them for a lower and an upper threshold criterion, respectively. Also, in order to focus on situations were a continuous flow is hold down by some jam situation, we focus on 8am to 8pm. In order to exclude situations too short to be sure whether they are false positives, we search for situations were the above criteria hold for at least two consecutive time bins. And finally, we take also the 2h mean speed and mean volume into consideration. We choose outlier situations from 9 Tuesdays in September and October 2019 and two cities from the core competition (Berlin and Istanbul, 100 tests for each city). This should give us situations without temporal shift with respect to the training data, neither due to Covid nor due to summer holiday.

In summary, we search and select pixels (just one speed and volume channel) using the following conjunctive filter criteria:

1. volume and speed quantiles (volume above 90% quantile, speed below 5% quantile)and volume threshold (volume above 5)
2. time between 8 A.M. 8P.M.
3. outlier duration (at least two consecutive time bins)
4. outlier mean volume and speed (outlier mean volume above 1.5 times 2h mean, outlier mean speed above 0.7 times 2h speed mean)

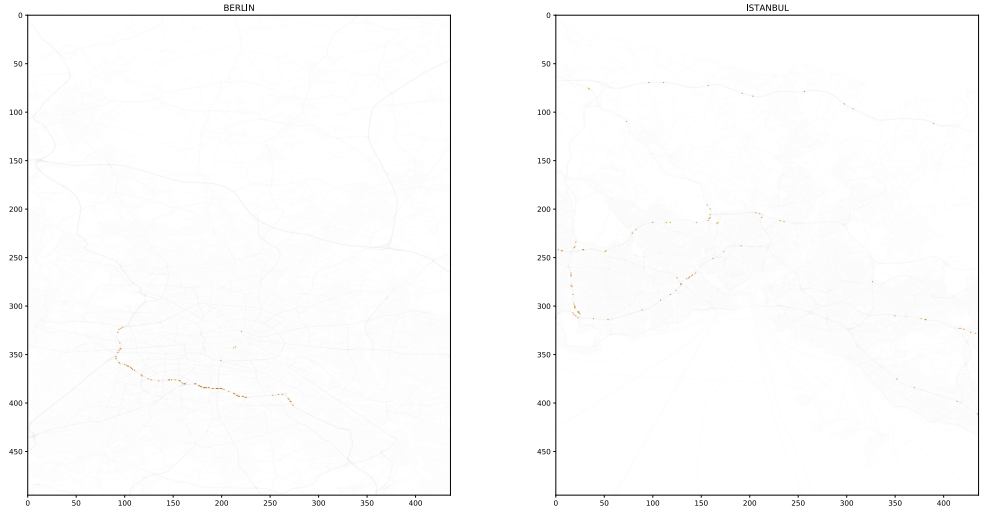
These criteria might be highly redundant and we did not check for a minimal set. These criteria are supposed to give us situations where volume is high and speed is low in contrast to normal situation as reflected by full-day quantiles and 2h means and which are robust (two consecutive positives and excluding sparse situations during night time). Formally, an outlier is a quintuple (row, column, heading, start time of day, duration).

For each test, we keep a mask that evaluates MSE on one pixel and two channels, the volume and speed channels of the outlier heading.

C.3. Test Slot Distribution

The procedure just described was applied on 9 Tuesdays in September/October 2019 before Covid temporal shift. Due to data scarcity, the above criteria were not applicable to Chicago and Melbourne from the core competition, so we excluded them as we were interested in the qualitative analysis of the situations. We randomly sample 100 outliers situations per city from 559 (Berlin) and 1266 (Istanbul). The other two cities of the core competitions turn out not to match the pre-conditions of the heuristics and are not included in the Outlier Special Prize. We generate tests such that the first outlier time bin is the last bin of the 1 hour test input window.

The spatial and temporal distribution of the slots are shown in Figures 23–24, respectively. As expected, the spatial distribution is concentrated along main arteries with continuous flow, and the temporal distribution reflects the morning and afternoon peak hours.

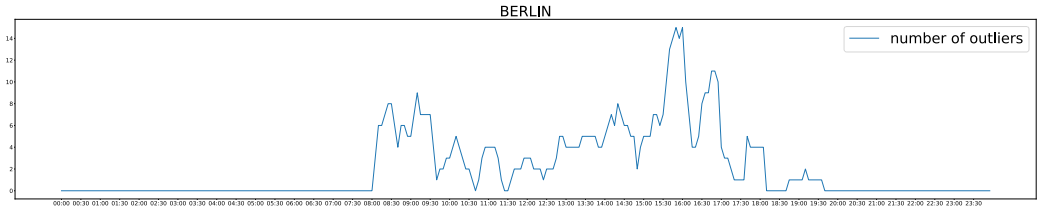


(a) Berlin (b) Istanbul

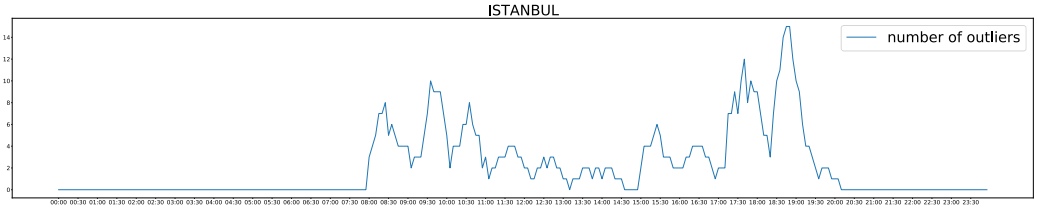
Figure 23: Spatial Distribution of Special Prize Tests.

C.4. Outlier Special Prize Submissions and Leaderboard

We invited all Summit/Symposium Participants of *Traffic4cast* 2021 to re-run their models on a new test set for the *Traffic4cast* 2021 Outlier Special Prize, and from 8 summit participants invited, 7 participated with a total of 11 submissions. 4 participants submitted the solution from re-running their core competition best model, but also submitted a second solution:



(a) Berlin



(b) Istanbul

Figure 24: Temporal Distribution of Special Prize Tests.

oahciy_v1 is obtained by exactly the same models that as for the core competition (Lu, 2021).

oahciy_v2 is obtained by further fine-tuning the models for 1 epoch on BERLIN and ISTANBUL data only.]

ai4ex_36 Epoch=36: This is exactly the submissions for the competition (plain vanilla) (Bojesomo et al., 2021).

ai4ex_43 Epoch=43: further training the model.

nina_orig the predictions of the original model (Unet++ patch-based prediction with 100x100 patch and stride s=10 as explained in (Wiedemann and Raubal, 2021)).

nina_special trained the original model further on the two cities separately. In detail, they loaded the model weights, then trained for 500 further epochs on Berlin, and predicted the values for the new Berlin test set. Same for Istanbul (starting again from the weights of the original submission).

sungbin_1 is from best run models on core task (Choi, 2021).

sungbin_2 is from best run models on extended task with different training methods which had not used in the core task (data augmentation method: input image flipping).

The other 3 submissions are jaysantokhi (Santokhi et al., 2021), GraphUNet_luca (Hermes et al., 2022) and Bo (Wang et al., 2021a).

For each test, we keep a mask that evaluates MSE on one pixel and two channels, the volume and speed channels of the outlier heading, see Figure 25. Hence, the quantitative evaluation takes place on much less data than the full city (as in Figure 26): We observe that oahciy_v2 (Lu, 2021), sungbin_1 (Choi, 2021) and oahciy_v1 (Lu, 2021) are very close on masked MSE. Surprisingly, sungbin_2 (Choi, 2021) based on the winning extended challenge

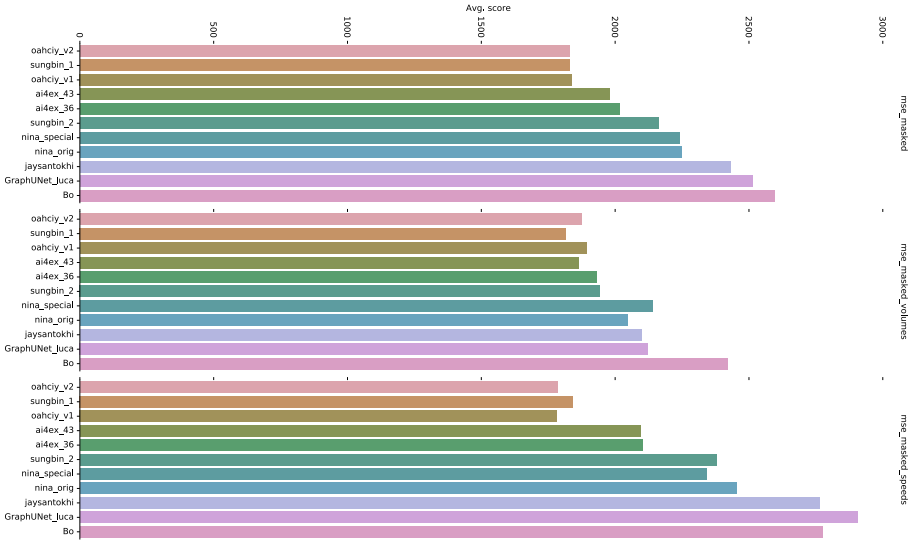


Figure 25: Leaderboard special prize (anomalies) based on masked MSE highlighting one cell and one heading.

is much poorer than sungbin_1, so it seems that sungbin_2 is too "conservative" in this setting where the best guess in most situations is predicting a normalization of the situation. The solutions ai4ex_43/aiex_36 (Bojesomo et al., 2021), nina_special/nina_orig (Wiedemann and Raubal, 2021), jaysantokhi (Santokhi et al., 2021), GraphUNet_luca (Hermes et al., 2022) and Bo (Wang et al., 2021a) are clearly beaten in the Special Prize challenge as well.

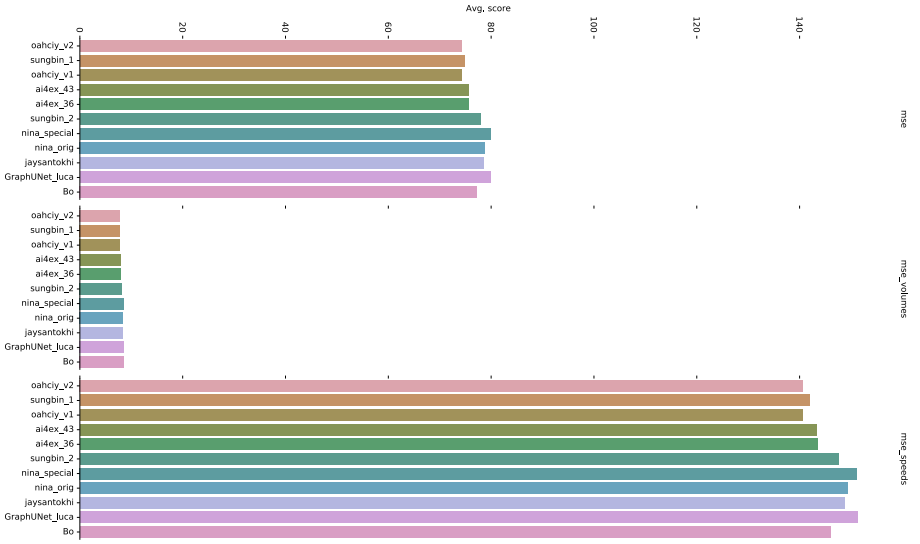


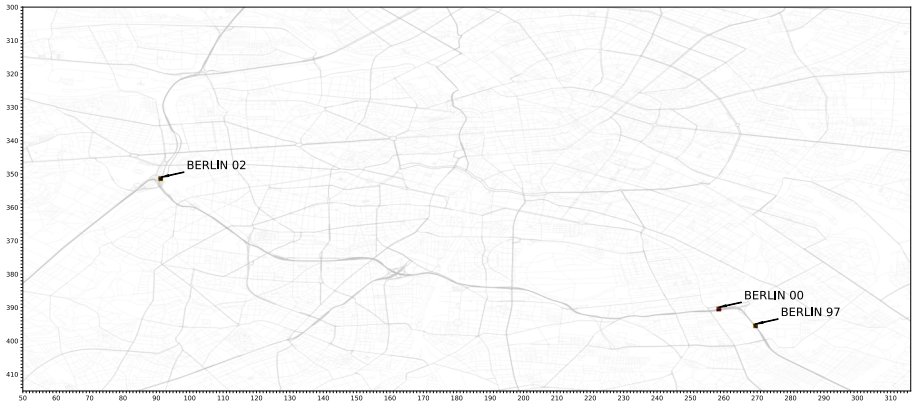
Figure 26: Unmasked MSE in special prize (anomalies) based on masked MSE.

As a sanity check, Figure 26 shows the unmasked MSE, which again shows similar level of MSE overall and for volume and speed separately as in the core competition. We see here the same top-3 submissions as in the core competition (oahciy_v1 marginally better than

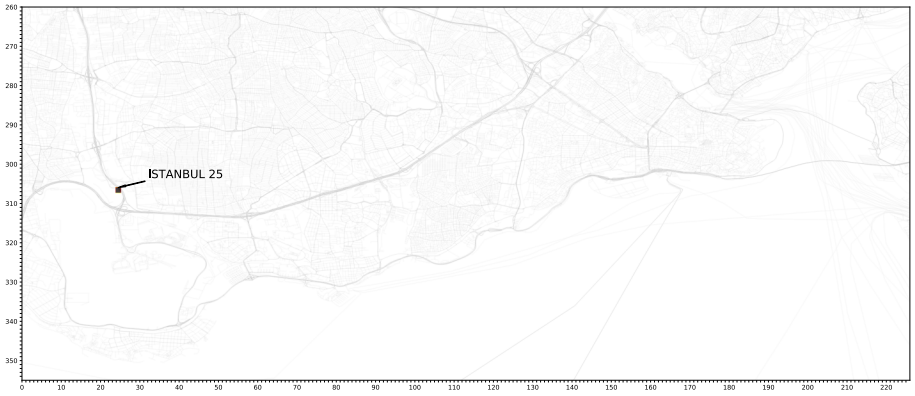
oahciy_v2, Sungbin_1, ai4ex_36, the third prize winner did not participate in the Special Prize). Sungbin_2 trained for unseen cities is clearly inferior to Sungbin_1 in both evaluations.

C.5. Special Prize Qualitative Analysis

Out of the 200 tests (100 for Berlin and Istanbul each), we choose 4 sample situations (see the outlier locations in Figure 27 and describe the anomalies qualitatively:



(a) Berlin



(b) Istanbul

Figure 27: Sample situations

BERLIN 00 Figure 28 : the anomaly started 40 minutes before, stop and go during input hour, normalizing in prediction hour after 10 minutes. We see speed drops during the night and in the evening due to zero volume. The outlier is in the afternoon peak. Prominently, speed has gone down and volume has gone up. The prediction seems to go beyond the mean speed and volume in the input, approximating normalization to "free flow speed". The highlighted gray area seems to be only at the second half of the jam. However, there was already a partial resolution of the jam, hence our outlier detection heuristic detected two consecutive outlier ane we see only the second sampled here. The winner prediction (oahciy_v2 (Lu, 2021)) suggests a steady normalization of volume and speed.

BERLIN 02 Figure 29 unsteady flow during input hour, anomaly started 5 minutes before: peak just before prediction start, normalizing over 30 minutes

BERLIN 97 Figure 30: anomaly 15 minutes before, the jam does not fully resolve during the prediction horizon and we suspect multiple minor go and stops during the prediction horizon (there are two volume peeks in the prediction horizon with low speeds at the same time, which of course could be due measurement error), with speed going up only slowly at the end of the prediction horizon. In this case, the model again predicts a smooth normalization, but far too soon. If we look at the full day ground truth, however, we see that the jam resolved shortly after the prediction horizon, so in some sense the model anticipated that with one smoothed idealised guess, reflecting multiple scenarios only in a statistical sense.

ISTANBUL 25 Figure 31: slowdown started 30 minutes before, going back to normal over 30 minutes

Without a quantitative verification, Figures 28–31 seem to suggest there are 3 typical behaviours:

monotonic normalization The first type is a monotonic normalization to close to free flow speed as in the two examples shown. This is the normal red level we see before the anomaly starts on the left. And the dashed red line gradually creeps back to that level over the one hour prediction horizon left to the vertical now line. The same happens for volume (blue) which goes down when the jam resolves. However again, the prediction does not fully recover so the model probably learns from the training data to expect to expect not full resolution of the jam over the prediction horizon. They seem to provide a "smoothing" of the anomaly going back to normal. Clearly overestimating in Berlin 97, where the situation does not normalize during the prediction horizon.

monotonic towards mean of input The second class does predictions which are monotonic towards mean of input. The mean of the input is shown by the faint horizontal red line. So it looks as if in these cases only consider the local effects in the input.

static The third class does a jump and stay prediction. Some models jump to the mean speed of the input as in the example shown on the right-hand side. Other models of the third class jump even to something close to free flow speed.

All models seem to predict pretty high volumes from what we would expect as the non-jammed normal volume from the input hours in the left half of the plots.

In summary, the best models predict smoothed version of a jam resolution (Berlin 00), underestimating speed and overestimating density; models are fooled in case jam does not resolve (Berlin 97), MSE makes prediction blurred towards the mean in the data, never predicting a rare scenario such as jam resolving more or less quickly than in expectation; MSE for volume and speed is at the same level.

We do not provide a thorough and systematic exploration of this classification, but we think it still illustrates some shortcomings of the current task formulation discussed in the main text.

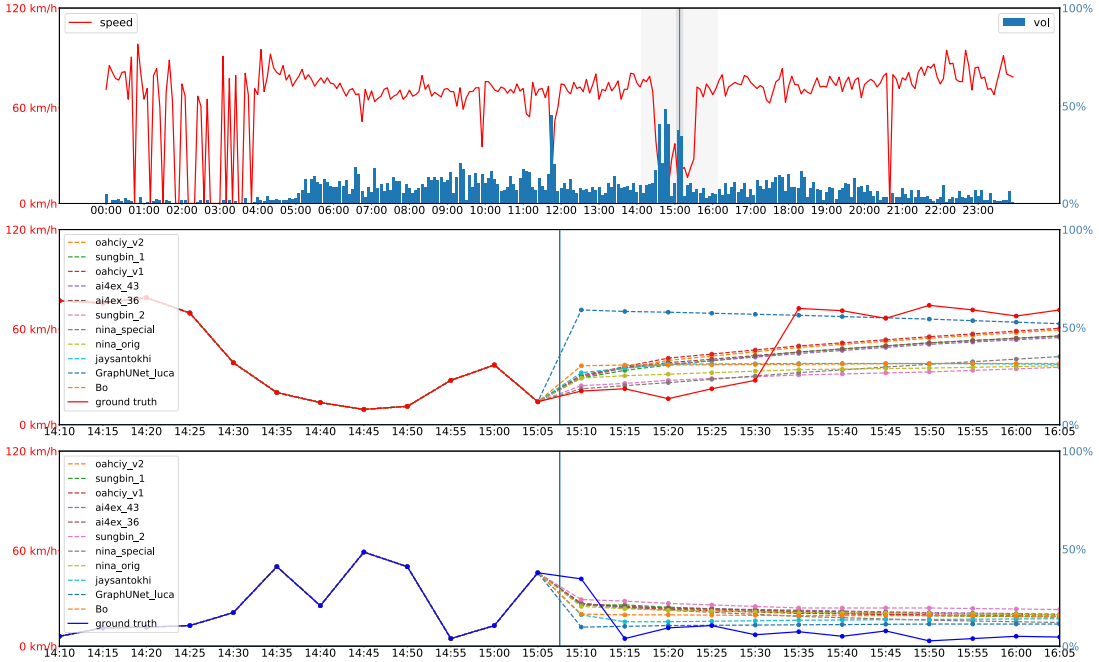


Figure 28: Predictions Berlin 00. Top: Daytime curve of ground truth data for 288 bins from midnight to midnight with speeds (red curve) and volumes blue bars. Outlier in darker gray area with input and prediction hour in light gray separated by a blue vertical line. Middle, bottom: The dashed lines show the speed and volume predictions of the different submissions compared to ground truth (solid lines); the vertical blue line separates input from output in time.

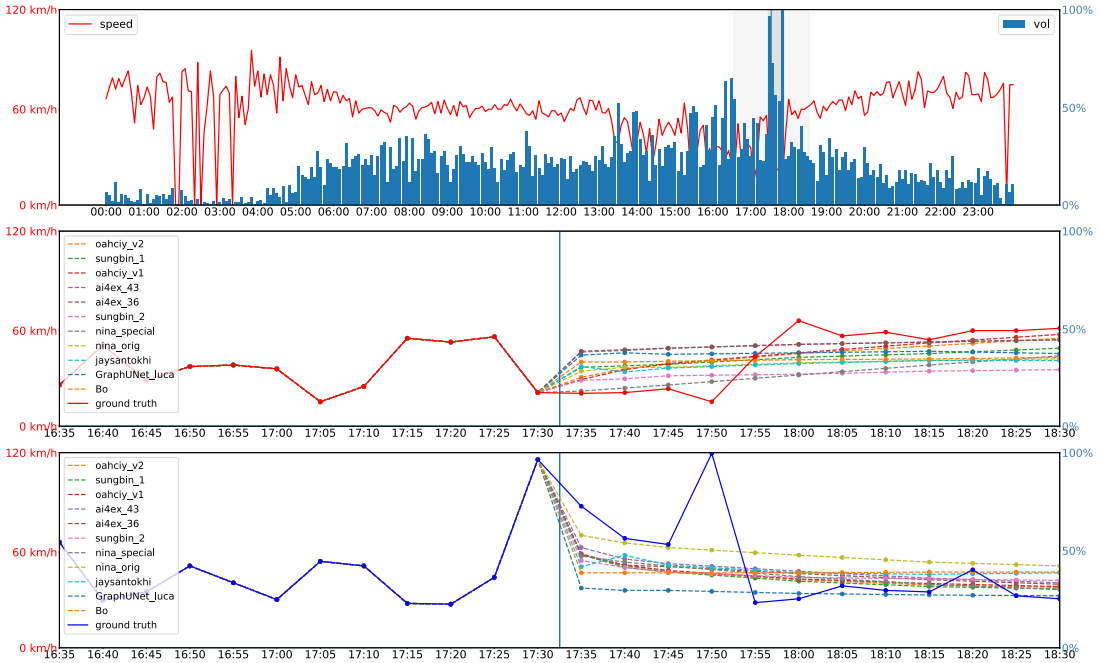


Figure 29: Predictions Berlin 02

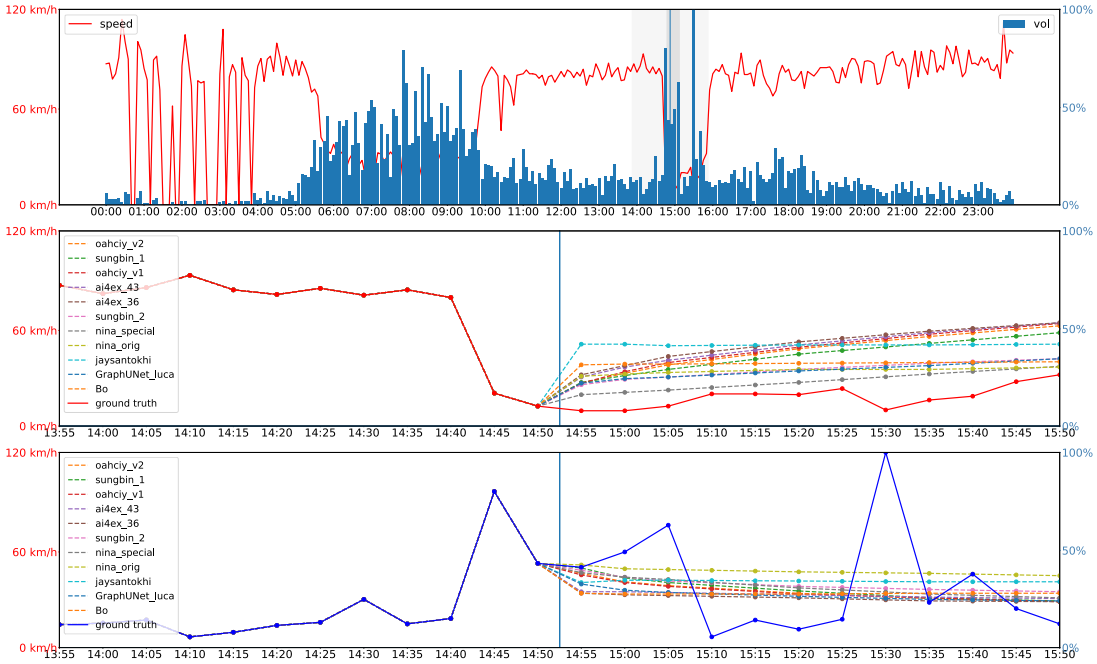


Figure 30: Predictions Berlin 97

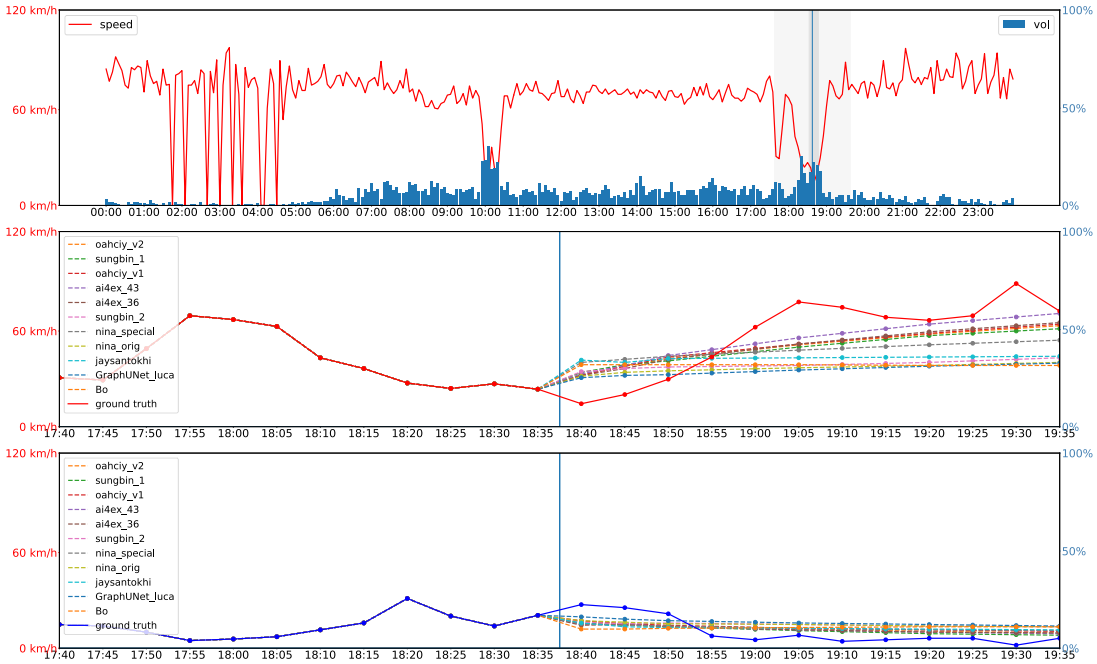


Figure 31: Predictions Istanbul 25

Appendix D. Leaderboards Core and Extended Competitions

In this appendix, we highlight some aspects of the leaderboard to highlight some features of MSE evaluation in the *Traffic4cast* setting. The code used to generate them and more plots can be found in (Eichenberger and Neun, 2021).

Figure 32 shows the dominance of speed channels in both competitions.

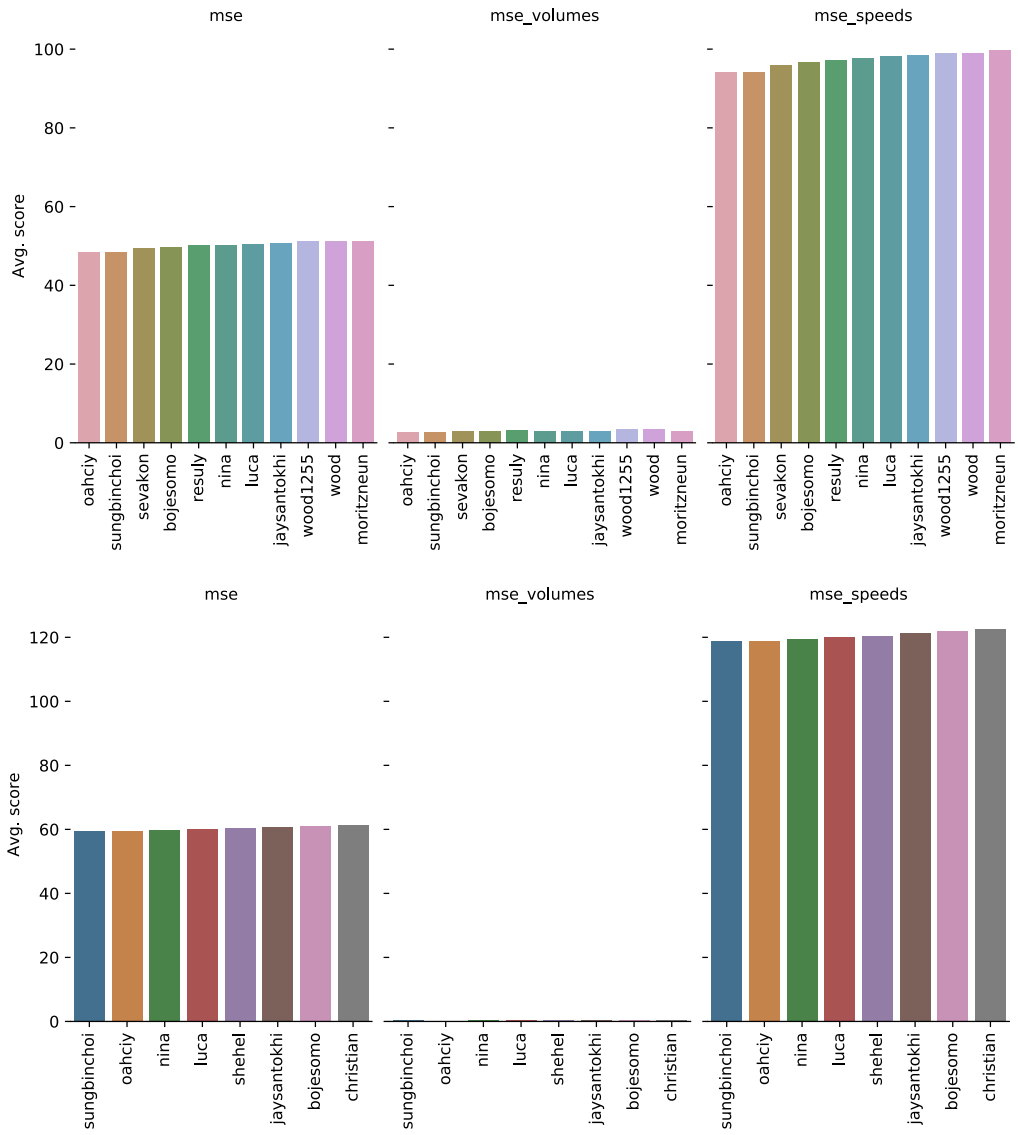


Figure 32: MSE Volume vs. speed bias: MSE over all channels and MSE on volumes and speeds separately for core competition (temporal shift, top) and extended competition (spatio-temporal shift, bottom). MSE is the average of volumes and speed MSE by definition.