Supplement: A baseline method for nowcasting count data

2025-07-16

# Supplemental methods

## Mathematical model

Our approach is based on the reference model for the COVID-19 hospitalisation nowcast challenge in Germany in 2021 and 2022 (1). In slight variation of the chain ladder method (2), it uses preliminary case counts and empirical delay distributions to estimate yet-to-be-observed cases. Probabilistic nowcasts are generated using a negative binomial model with means from the point nowcast and dispersion parameters estimated from past nowcast errors. Below, we describe the mathematical details of each component of the model, starting with a definition of the notation used throughout. In our description we assume that time steps correspond to days, but they could also be weeks, months or any other unit.

See the *Default Settings* section in the main text methods for a description of the default behaviour of the method within the baselinenowcast package.

### Notation

We denote by the number of cases occurring on day which appear in the dataset with a delay of days. For example, a delay means that a case occurring on day arrived in the dataset on day . We only consider cases reported within a maximum delay . The number of cases reporting for time with a delay of at most can be written as:

A special case of this is the “final” number of reported cases at time , denoted by

For delays we moreover define the notation

representing the number of cases still missing after days of delay. In the following we use uppercase letters () for random variables, lower case () for the corresponding observations, and hats () for estimated / imputed values. The matrix with entries is referred to as the *reporting matrix*:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

In the case where corresponds to the present date, all entries with have yet to be observed and are thus still missing. As the available entries at its bottom form a triangle, this incomplete reporting matrix is referred to as the *reporting triangle*.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### Pre-processing of the reporting triangle

All of the following steps require that the reporting triangle only has non-negative entries. In practice this is not necessarily the case. For instance, if the reporting triangle has been computed from increments in subsequent data snapshots, occasional downward corrections due to data entrance issues can cause negative entries. We therefore apply a pre-processing step to re-distribute negative entries across neighbouring cells with positive entries.

### Delay distribution estimation

In this section we detail how we estimate the reporting delay distribution based on the last rows of a reporting matrix or triangle. See the Methods: Overview section in the main text for more details on the package default for the number of reference times used for delay estimation.

#### Estimating the delay distribution from a reporting matrix

If a complete reporting matrix is available, estimating the discrete-time delay distribution is straightforward. Using the last rows of the reporting matrix , we compute

which is simply the relative frequency of a delay of days among all cases in the reporting matrix.

#### Estimating the delay from a reporting triangle

In the case where is the present day, such that only a reporting triangle with missing entries is available, the estimator from [Equation 2](#eq-pi_d) can only be evaluated after discarding all data from the last time points. In order to use these partial observations, we use a different representation of the delay distribution via terms of the form

for . Here, in analogy to [Equation 1](#eq-Xltd) we write

The can be estimated via

and translated to estimates via the recursion

subject to the constrain that .

We note that this method is equivalent to the so-called chainladder method (2), adapted to our notation in terms of reporting triangles (rather than *development triangles* as used in accounting).

### Point nowcast generation

We now address the computation of a point nowcast, i.e., expected total daily case numbers . These are based on the reporting triangle, more specifically the preliminary totals , and the estimated delay distribution, . In the following we will denote the current day as the nowcast date, and the days as the reference dates. The difference will be called the *horizon*.

An intuitive approach, used in (1) and the standard chain ladder technique, is to simply inflate the current total for a reference date by the inverse of the respective probability of observation up to time ,

This, however, is not well-behaved if no cases at have been observed yet, i.e., . Then is likewise zero, which yields problems in our uncertainty quantification method (see next section). Motivated by a Bayesian argument (see [Zero-handling](#zero-handling-approximation) below) we therefore use the expression

instead. This yields essentially identical results for large , but produces positive even for preliminary zero values .

For our uncertainty quantification scheme we require not only estimated totals , but all entries of a point nowcast matrix. For these are obtained as

### Uncertainty quantification

To estimate the uncertainty in the nowcasts, we use the nowcast errors from past nowcasting time points. See the Methods: Overview section for more details on the default settings used in the package to define the number of past nowcasting time points used.

#### Generation of retrospective reporting triangles

We first obtain “vintage’’ reporting triangles of the raw reporting triangle (i.e., before pre-processing) to replicate the data which would have been available as of times , i.e., the last days on which nowcasts could have been generated. This simply corresponds to the stepwise omission of all entries with , which for each is a diagonal from the bottom left to the top right. The same pre-processing step as in Section [Preprocessing of the reporting triangle](#preprocessing) is applied to each vintage reporting triangle.

#### Generation of retrospective point nowcast matrices

For each of the vintage reporting triangles, i.e., , we apply the method described above to estimate a delay distribution and generate a point nowcast matrix. To indicate the data version on which it is based, its entries are denoted by

for and .

Note that estimation is again based on the last rows of the respective reporting triangle, which must consequently contain at least rows in total.

#### Fit a negative binomial observation model to past nowcast errors (per nowcast horizon)

A point nowcast based on the reporting triangle from time can also be written as

Only the second term has some associated uncertainty, while the first is already known at time . To quantify this uncertainty for given nowcast horizon , we assemble the and for . If all observations were complete, we would then estimate the overdispersion parameter of a negative binomial distribution

with independence assumed across the different . This, however, is not directly feasible as again some of the are not yet observed at time . We could discard these instances, but this would considerably reduce our number of available observations. We therefore use partial observations as available at time and assume

Here, the use of a constant dispersion parameter despite some of the values in the fitting procedure being yet incomplete is justified by the fact that the negative binomial distribution is closed to binomial subsampling, with the overdispersion parameter preserved. If equation [Equation 6](#eq-negbin) holds in combination with

we thus obtain [Equation 7](#eq-negbin2). Estimated dispersion parameters are obtained by maximum likelihood estimation.

### Probabilistic nowcast generation

Predictive distributions for are obtained by combining our point nowcasts with the estimated overdispersion parameters in a simple plug-in fashion. Specifically, we set

The predictive distribution for then results by shifting this distribution by the already known value .

### Zero-handling strategy

As mentioned in [Point nowcast generation](#point-nowcast-generation), we use a modified point nowcasts to deal with zero values in preliminary counts. We here motivate this approach from a Bayesian perspective. To this end we assume that

We are now interested in the conditional expectation

in this binomial subsampling problem. We will derive it under the improper prior distribution

For notational simplicity and readability for the following, we substitute , and and are thus looking for if with a discrete uniform prior on . This expectation can be written out as:

Applying Bayes Theorem we have

Because is a constant this simplifies to

Now substituting the probability mass function of the binomial distribution

we get

Plugging this into we get the following (omitting terms for , which are 0):

This is equivalent to

Both the numerator and the denominator are known convergent series, with solutions available in standard libraries like Mathematica (3). We then get

which corresponds to the corrected point estimate provided in equation [Equation 5](#eq-correction).

### Description of KIT simple nowcast implementation issue and revised nowcasts

When verifying the KIT simple nowcast implementation, we noticed that the code used to generate the KIT simple nowcasts in real-time contained an additional indexed element in the delay PMF for all counts observed beyond the maximum delay thus far. In the generation of retrospective nowcasts, these values were being handled as having been observed as 0s, when in fact they had yet to be observed in all of the retrospective nowcasts. This resulted in a comparison of a 0 to a value that in some cases is eventually observed, rather than beign excluded as they should have been. This has the impact of inflating the dispersion estimates, and may explain why the KIT simple nowcast method in Wolffram et al. (1) overcovers relative to the other methods. In the latest implementation in the RESPINOW Hub (4), the authors have removed the density beyond the maximum delay column from their implementation, and thus it no longer contains this issue.

In order to validate our baselinenowcast method against a revised version with the bug fixed, we regenerated the nowcasts using the pre-processed reporting triangle in the German Nowcast Hub repository (5) and generated retrospective nowcasts, referring to these in the main text and supplement as the KIT simple nowcast revised.

## Additional figures

### German Nowcast Hub validation using revised KIT simple nowcast

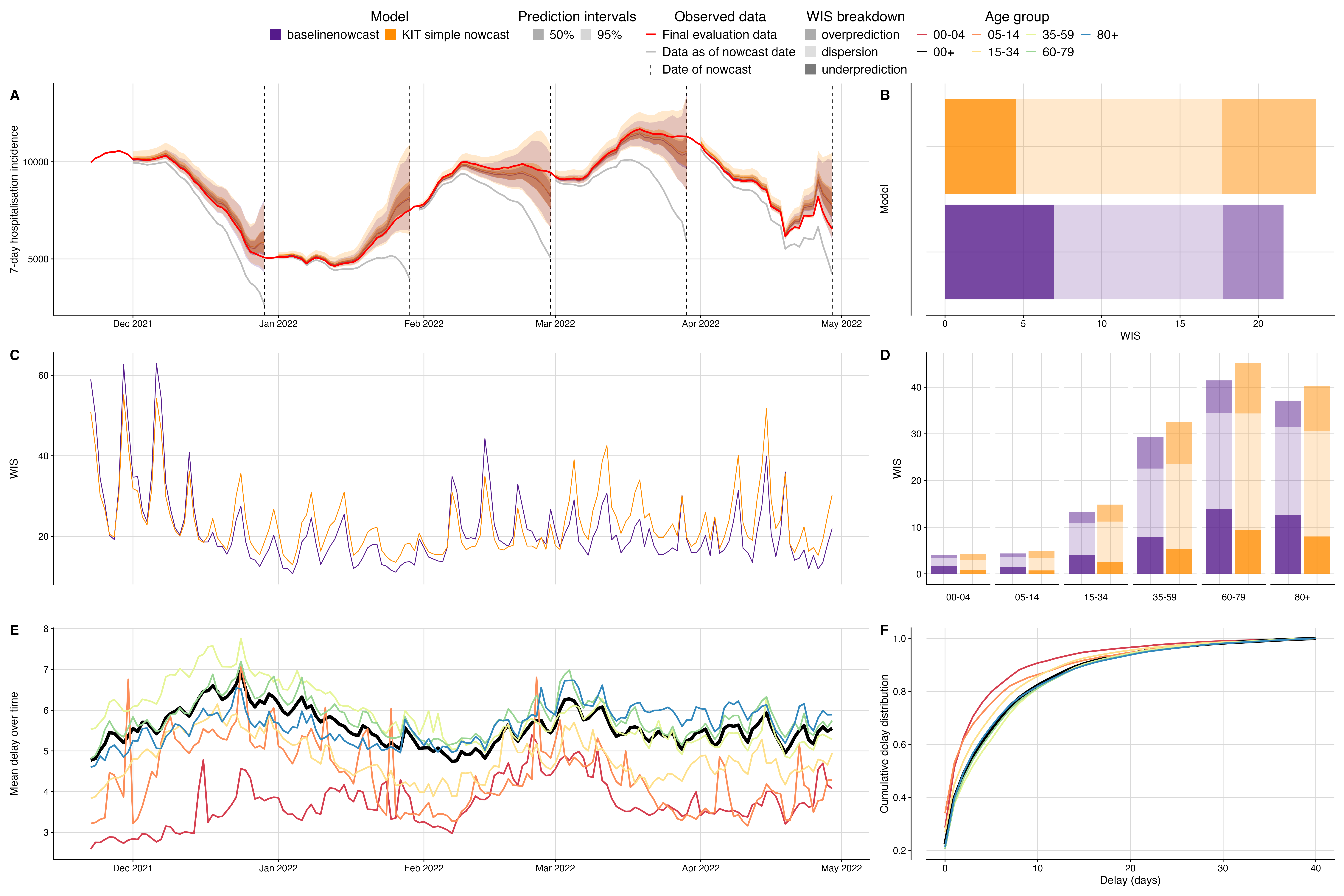


Fig. S1 Validation of baseline nowcasting model using German COVID-19 data compared to real-time KIT simple nowcast. A. Illustration of the nowcasting task from 5 nowcast dates throughout the study period. Nowcasts from our baseline model (purple) and the KIT simple nowcast model plotted against eventually observed values aggregated across all age groups, coloured by model. Observed data are shown as solid lines with gray indicating data available at the nowcast date and red indicating the final evaluation data. Dashed vertical lines indicate the date of the nowcasts. Shading indicates prediction intervals of each nowcast. Nowcasts are shown from horizons 0 to -28 days. B. Overall performance comparison between models aggregated across performance in individual age strata, with decomposed WIS (dispersion, overprediction, underprediction) displayed as stacked bar charts with different shading for direct comparison. C. Performance over time, displaying daily mean WIS scores summarised across age strata and 28 horizon days, coloured by model. D. Performance by age group, showing WIS scores across different age groups on the x-axis with grouped bars for both models and stacked shaded components displaying the decomposition of scores. E. Mean reporting delay over time, visualised as multiple coloured lines representing each age group, with the national average shown as a thicker black line. Mean reporting delays were calculated using the last 40 reference dates, corresponding to the maximum delay, for each nowcast date and age group. F: Delay distribution presented as cumulative distribution functions (CDFs), with overlaid curves for each age group and the national average highlighted as a thicker black line.

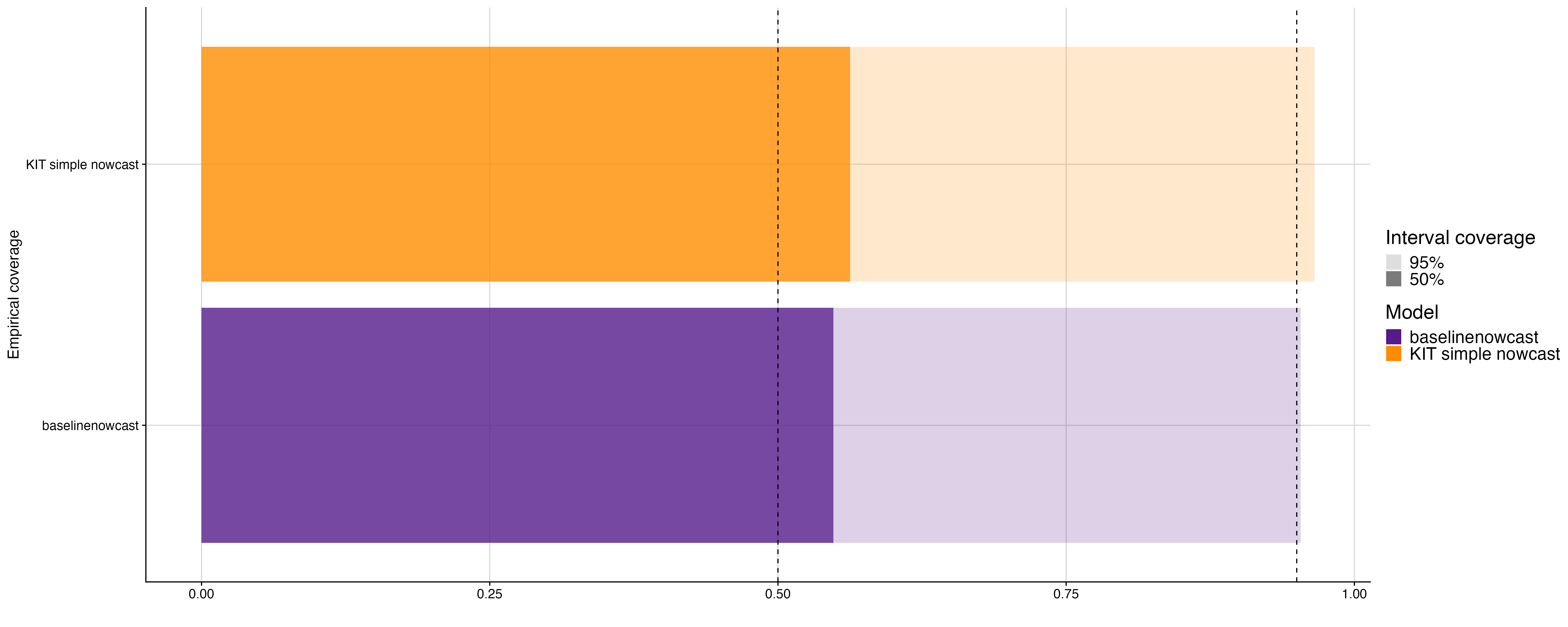


Fig. S2 Empirical coverage of baselinenowcast method and real-time KIT simple nowcast at 50% and 95% prediction intervals. Shading indicates prediction intervals, colours indicate model.

### German Nowcast Hub validation additional figures

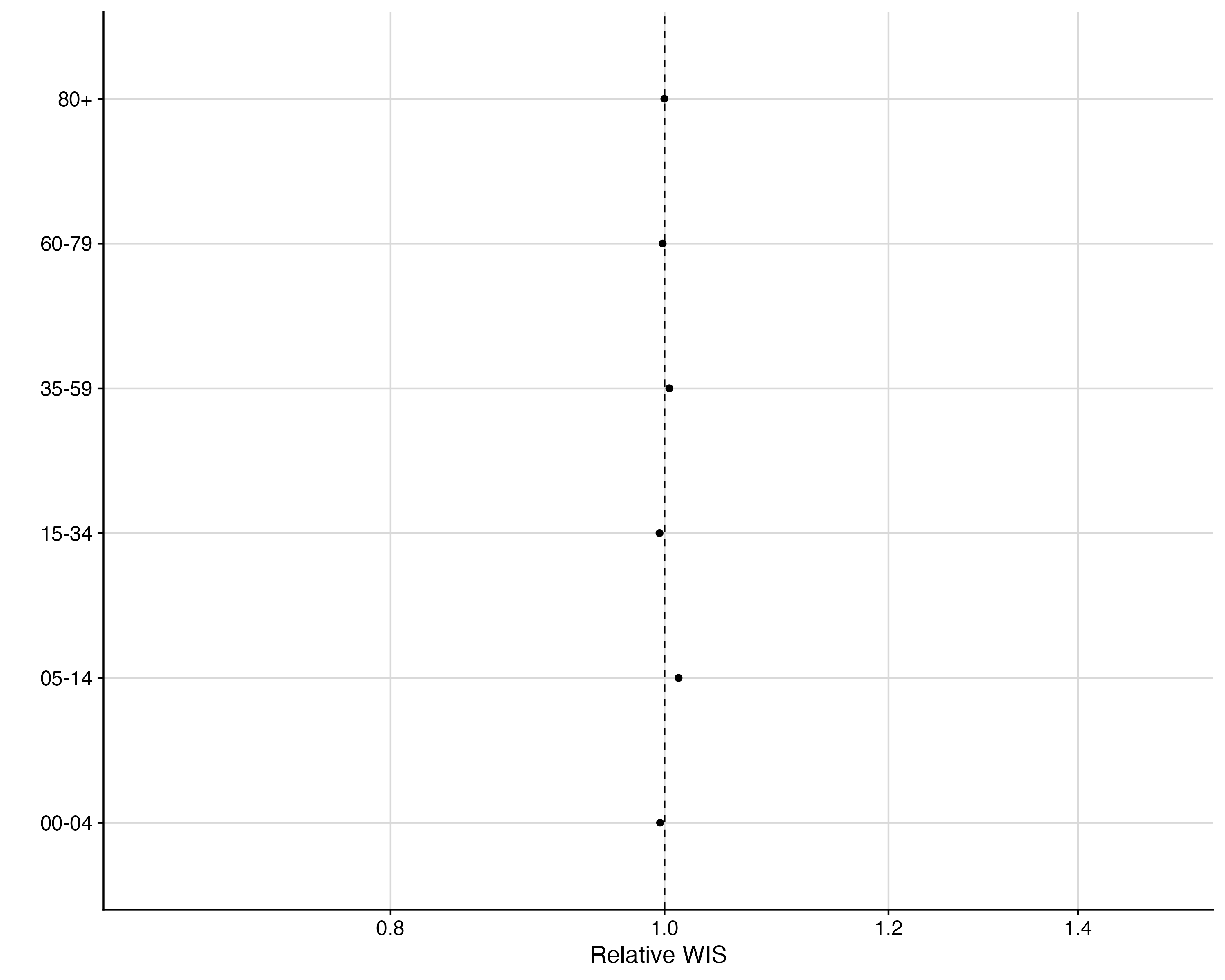


Fig. S3 Relative WIS by age group of baselinenowcast compared to KIT simple nowcast revised. Vertical line at 1.0 indicates parity with KIT simple nowcast revised.

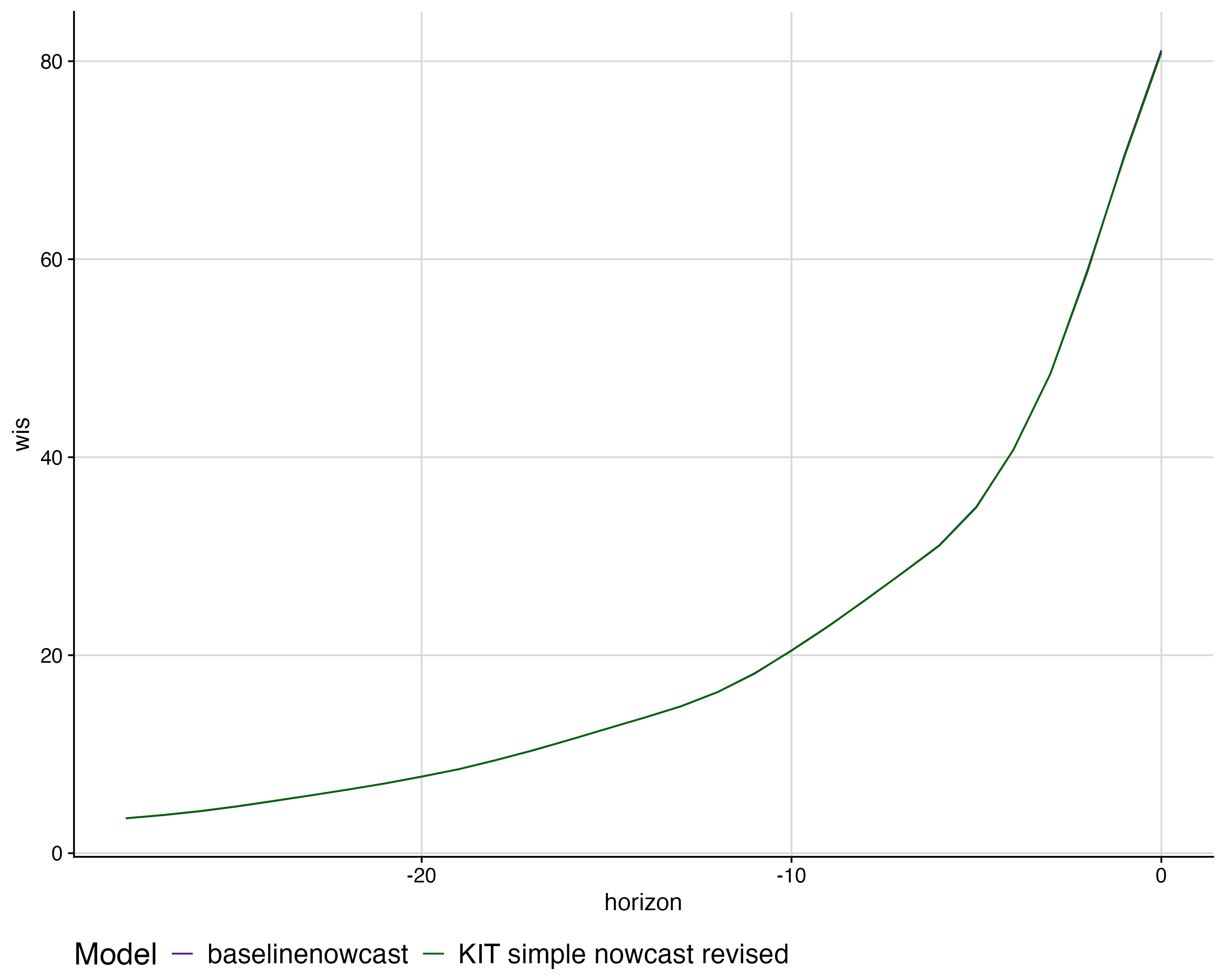


Fig. S4 Mean WIS by nowcast horizon for each model. colours indicate models.

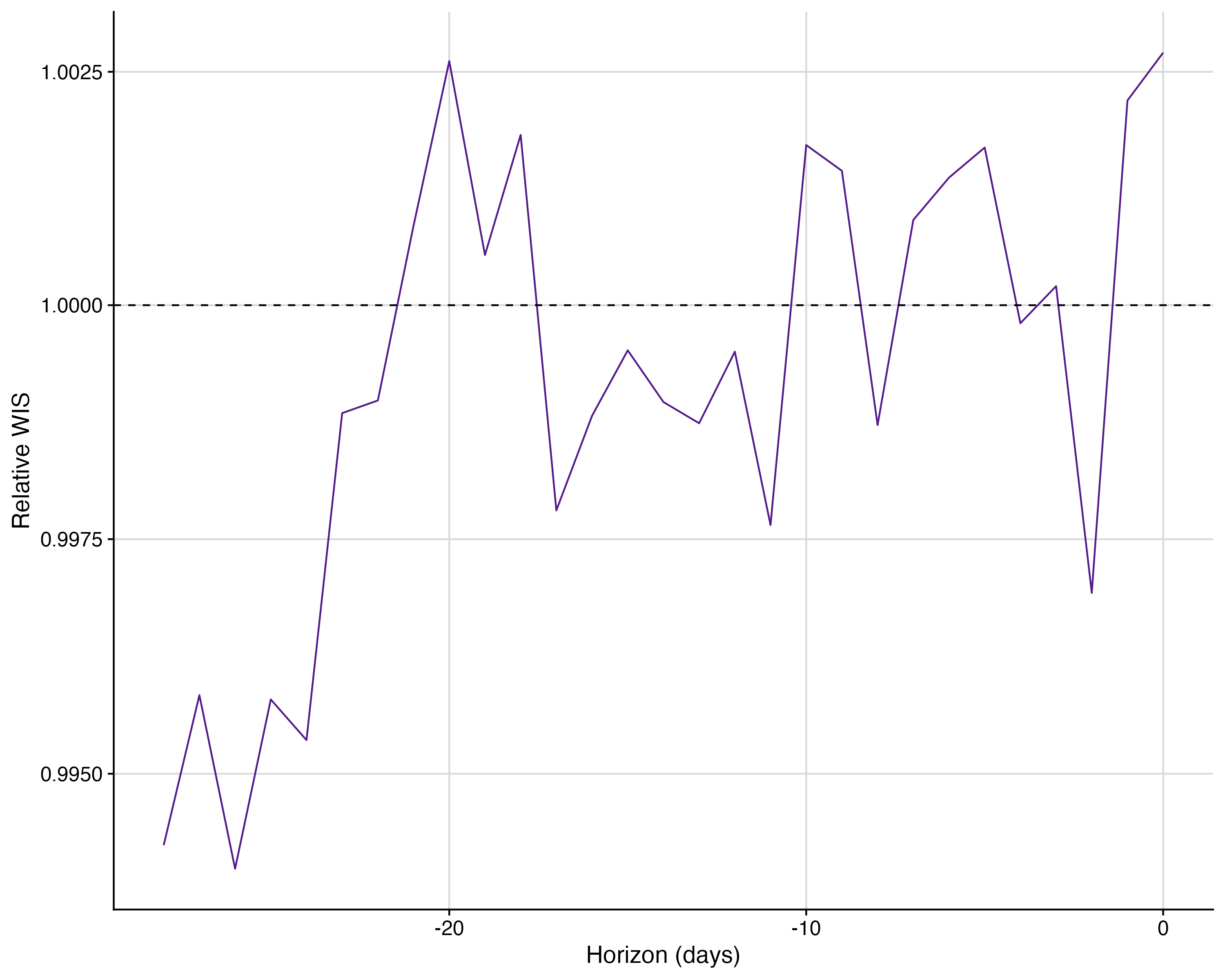


Fig. S5 Relative WIS of baselinenowcast horizon compared to KIT simple nowcast revised.

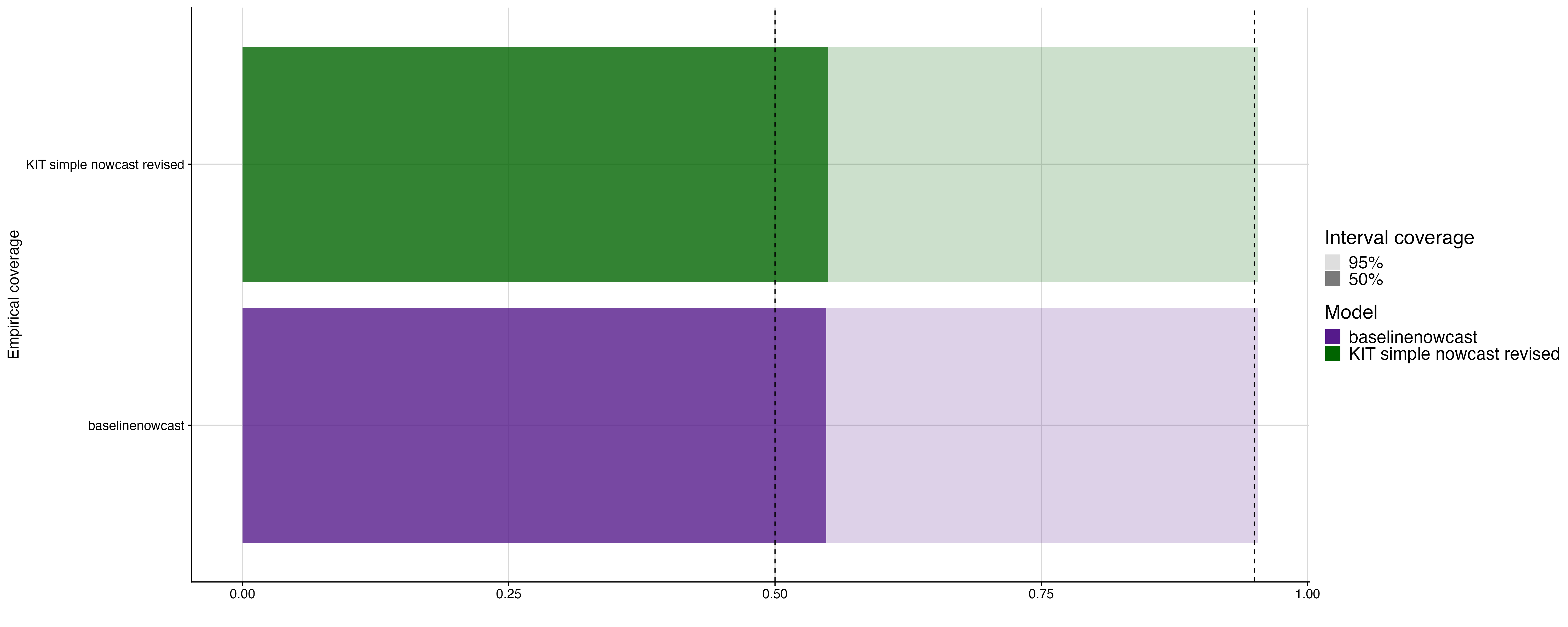


Fig. S6 Empirical coverage at 50% and 95% prediction intervals. Shading indicates prediction intervals, colour indicates models, vertical lines indicate the target prediction intervals.

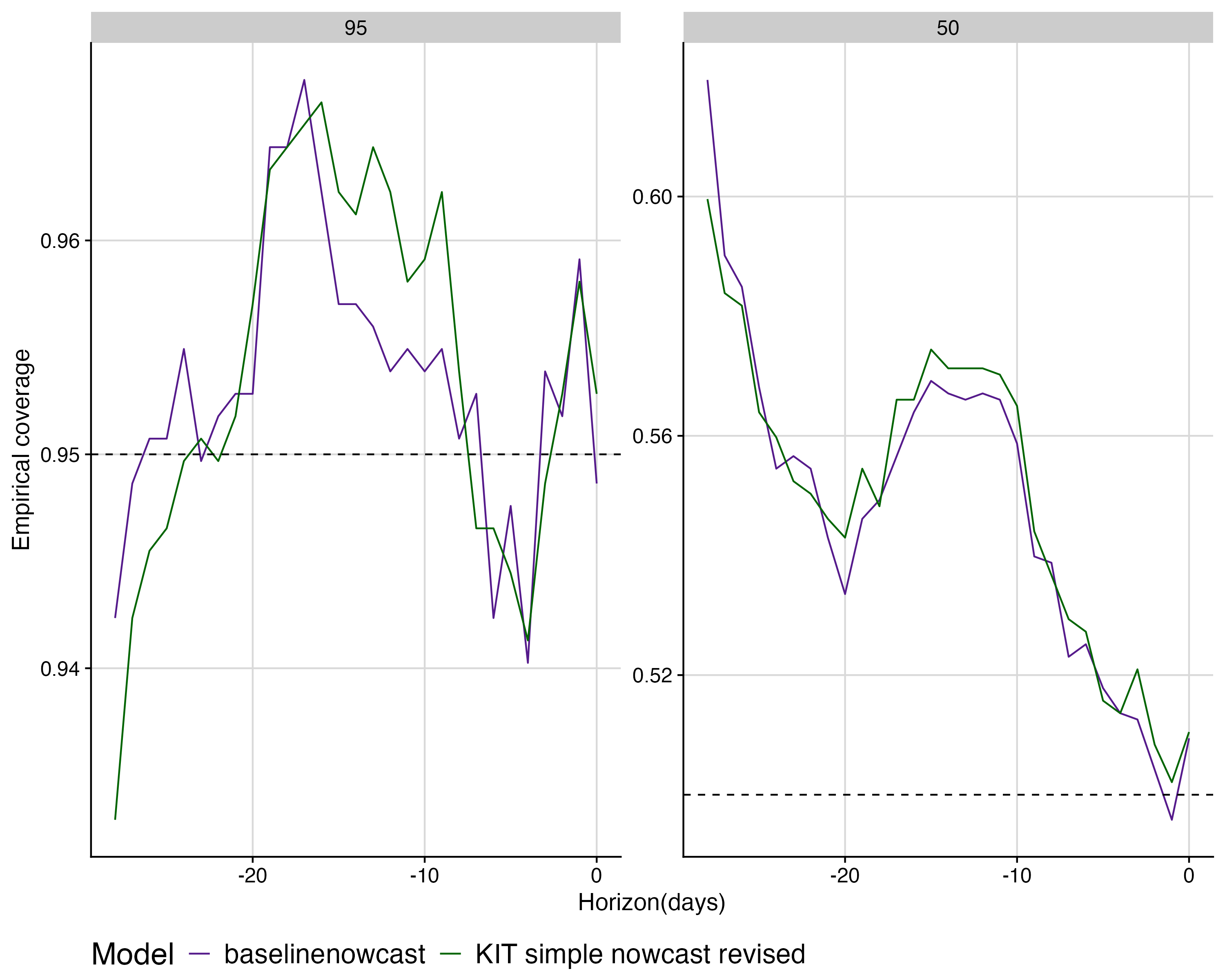


Fig.S7 Empirical coverage at 50% and 95% prediction intervals by horizon for each model. colours indicate models, horizontal lines indicate the target prediction intervals in each subplot.

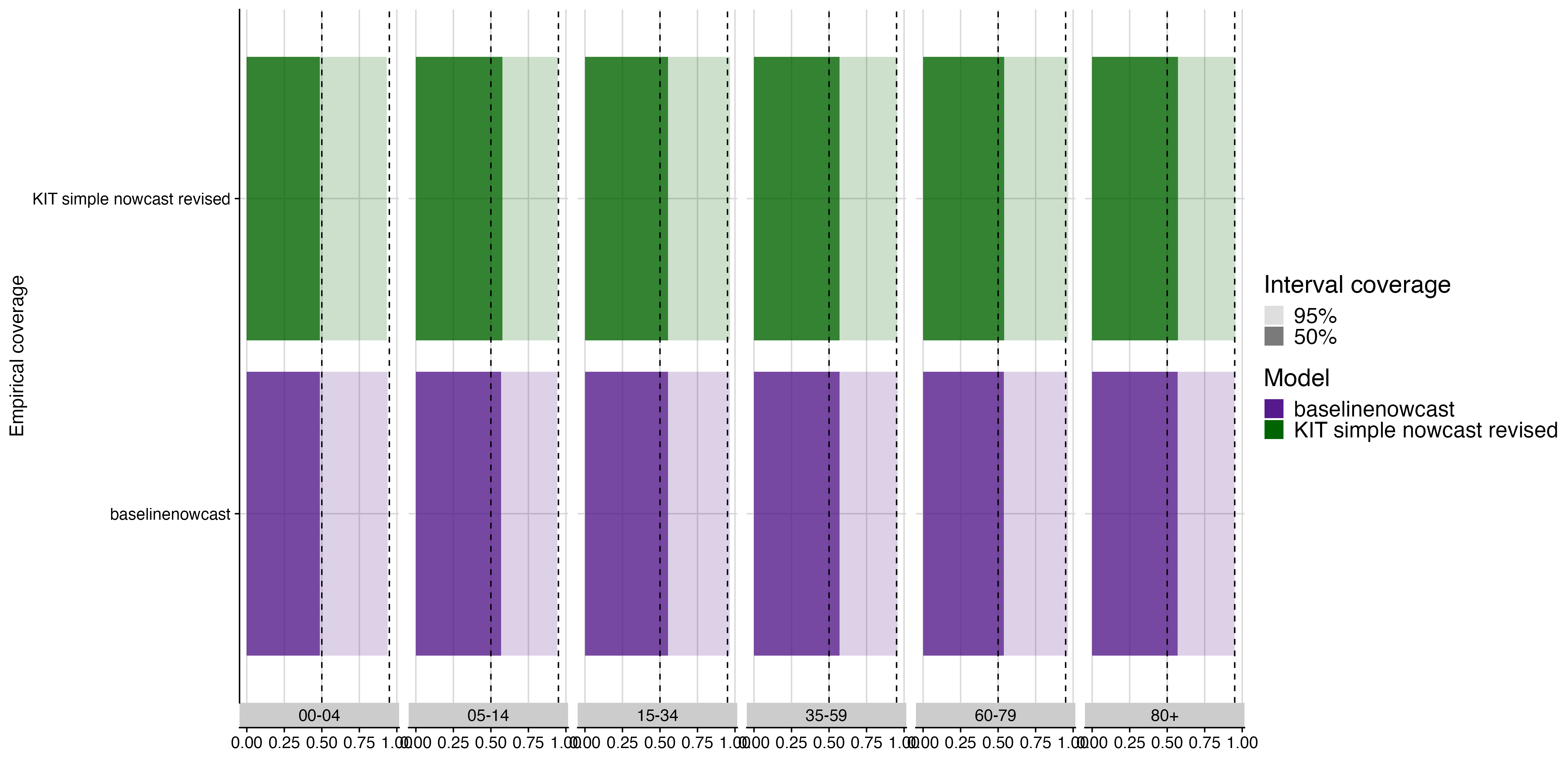


Fig S8. Empirical coverage at 50% and 95% prediction intervals by age group for each model. Shading indicates prediction intervals, colour indicates models, vertical lines indicate the target prediction intervals.

### Model permutation study

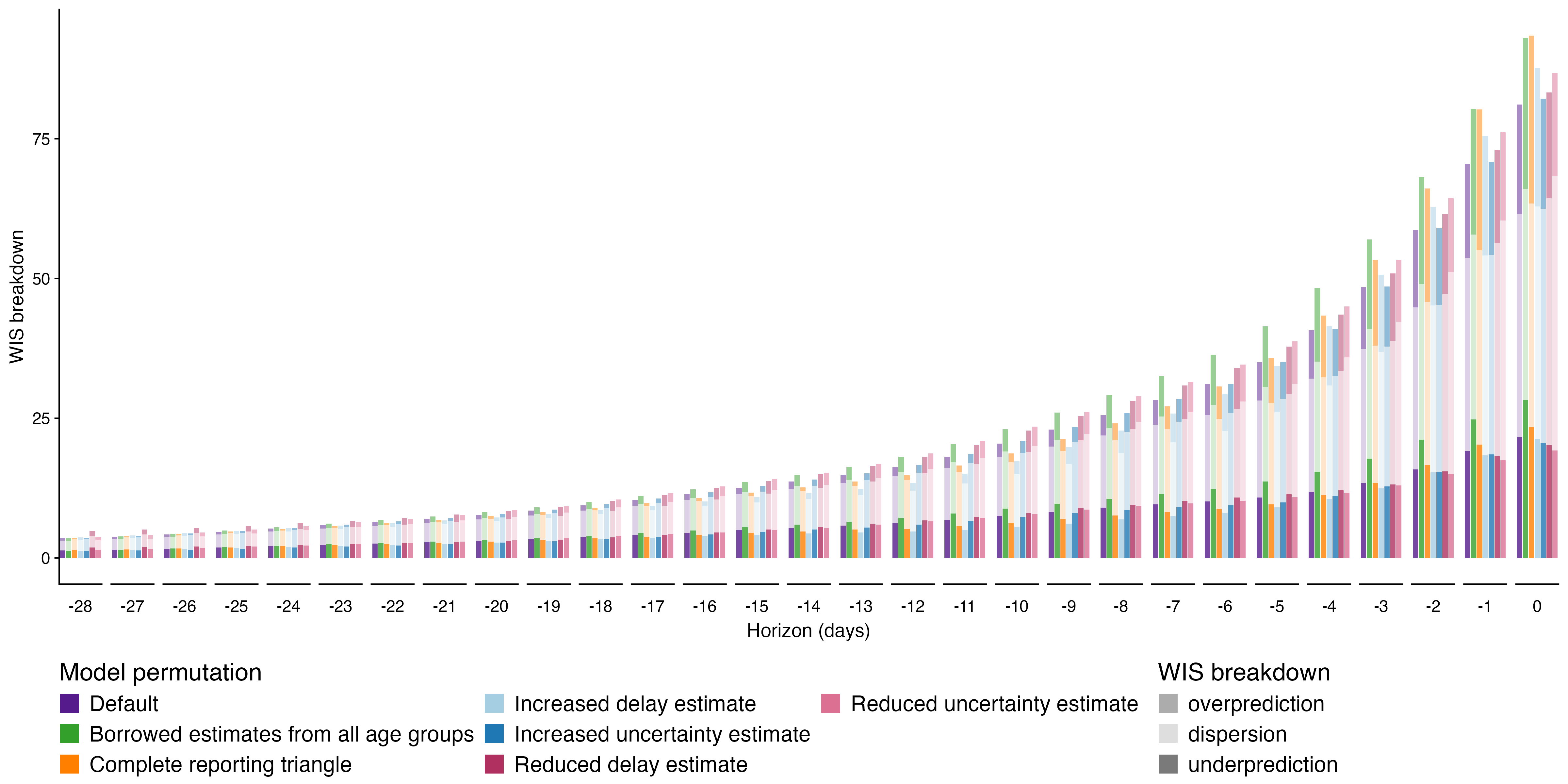


Fig S9. Absolute WIS by nowcast horizon for each model configuration. Colour indicates model configuration, shading indicates WIS decomposed by dispersion, overprediction, and underprediction.

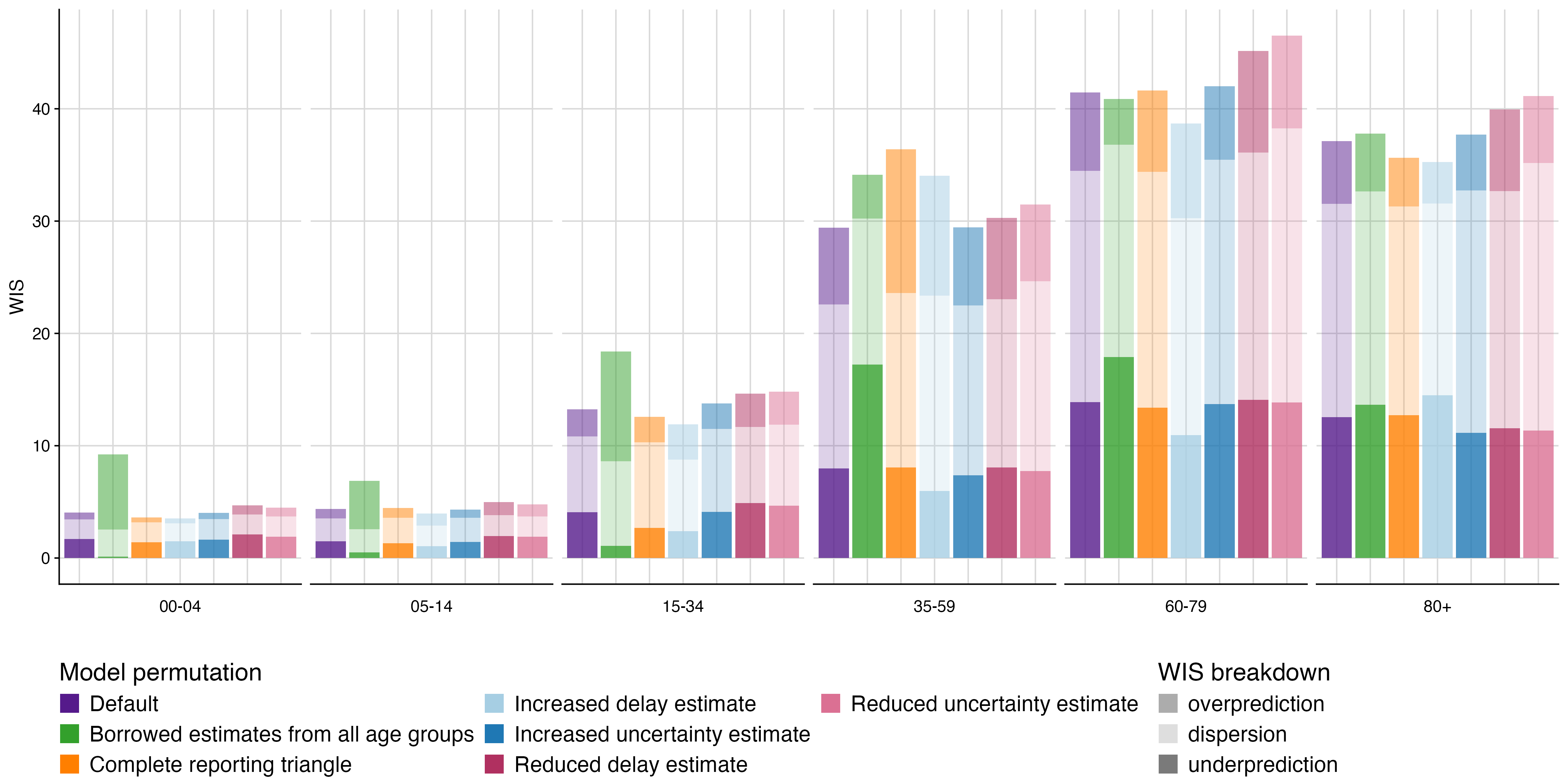


Fig S10. Absolute WIS by age group for each model configuration. Colour indicates model configuration, shading indicates WIS decomposed by dispersion, overprediction, and underprediction.



Fig S11. Absolute WIS over time (by week) for each model configuration. Colour indicates model configuration, shading indicates WIS decomposed by dispersion, overprediction, and underprediction.

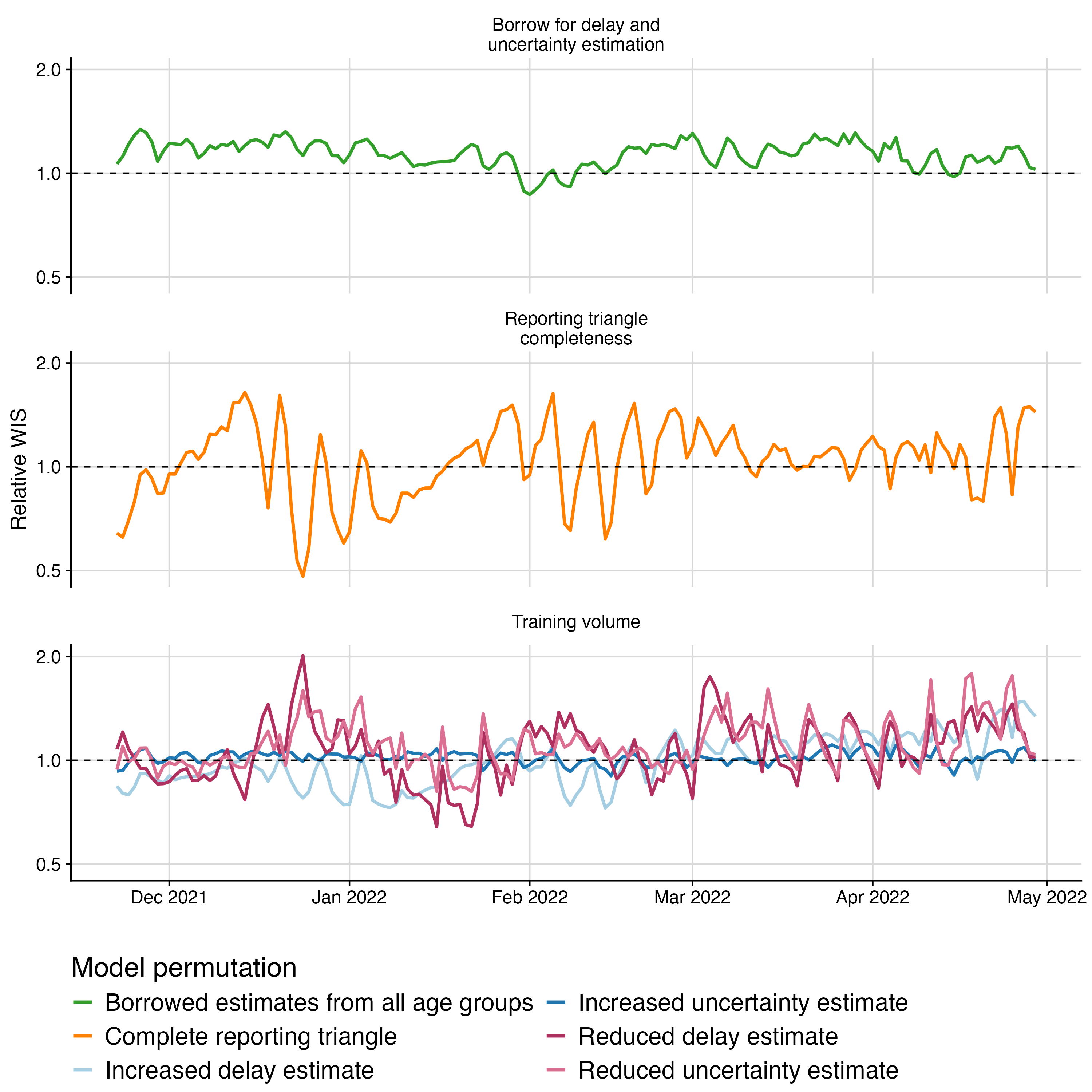


Fig S12. Relative WIS over time across all age groups for each model configuration. Colours indicate model configuration. Horizontal dashed line indicates parity with the default configuration.

### UKHSA norovirus surveillance model comparison

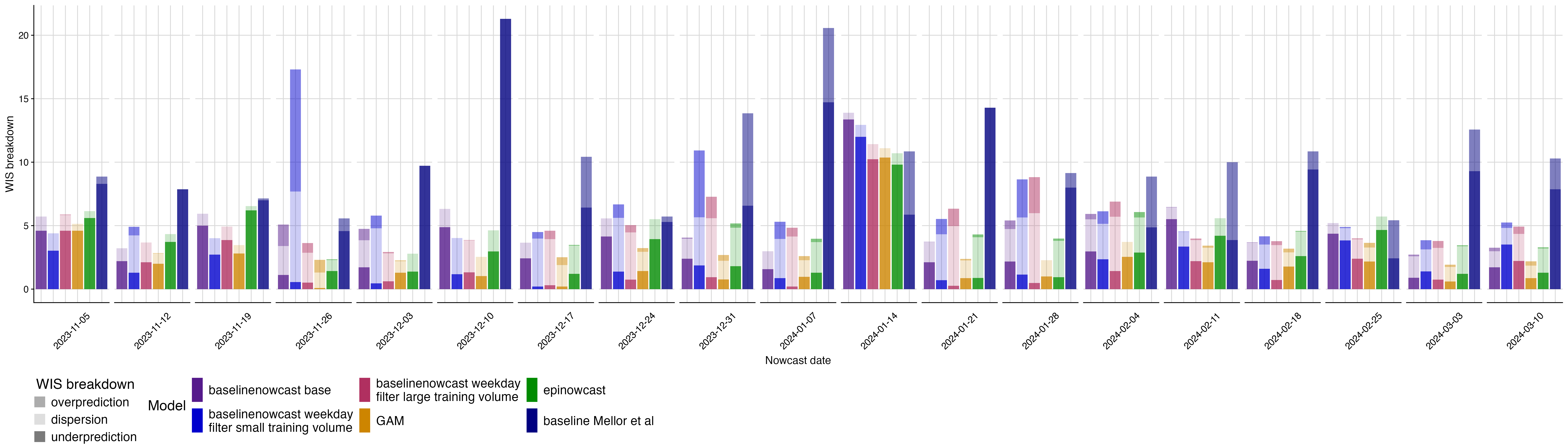


Fig S13. Absolute WIS over time (by week) for each model. colour indicates model, shading indicates WIS broken down by dispersion, overprediction, and underprediction.

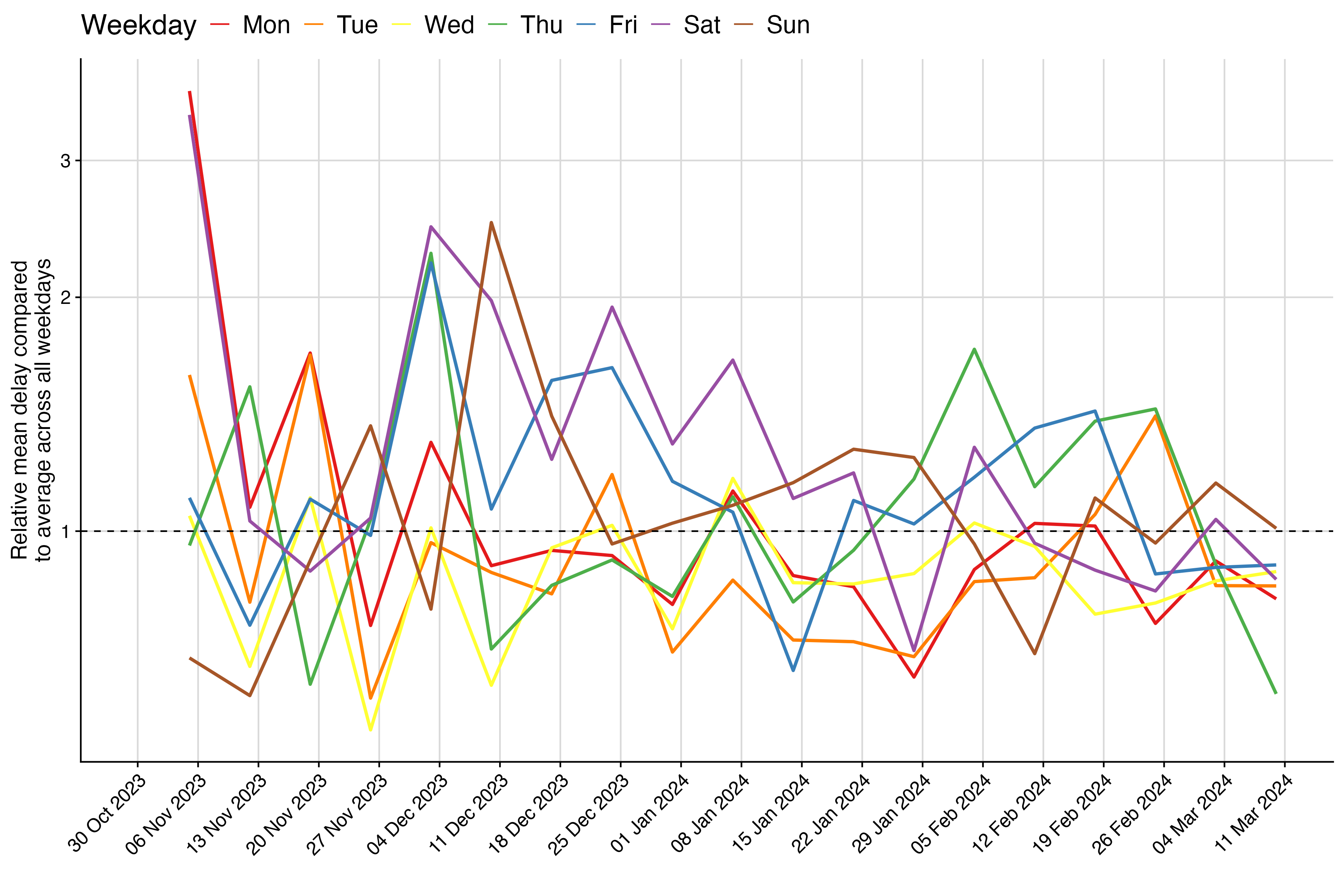


Fig S14. Relative mean delay over time by weekday. colours indicate weekday. Horizontal dashed line at 1 indicates parity with the delay across all weekdays.

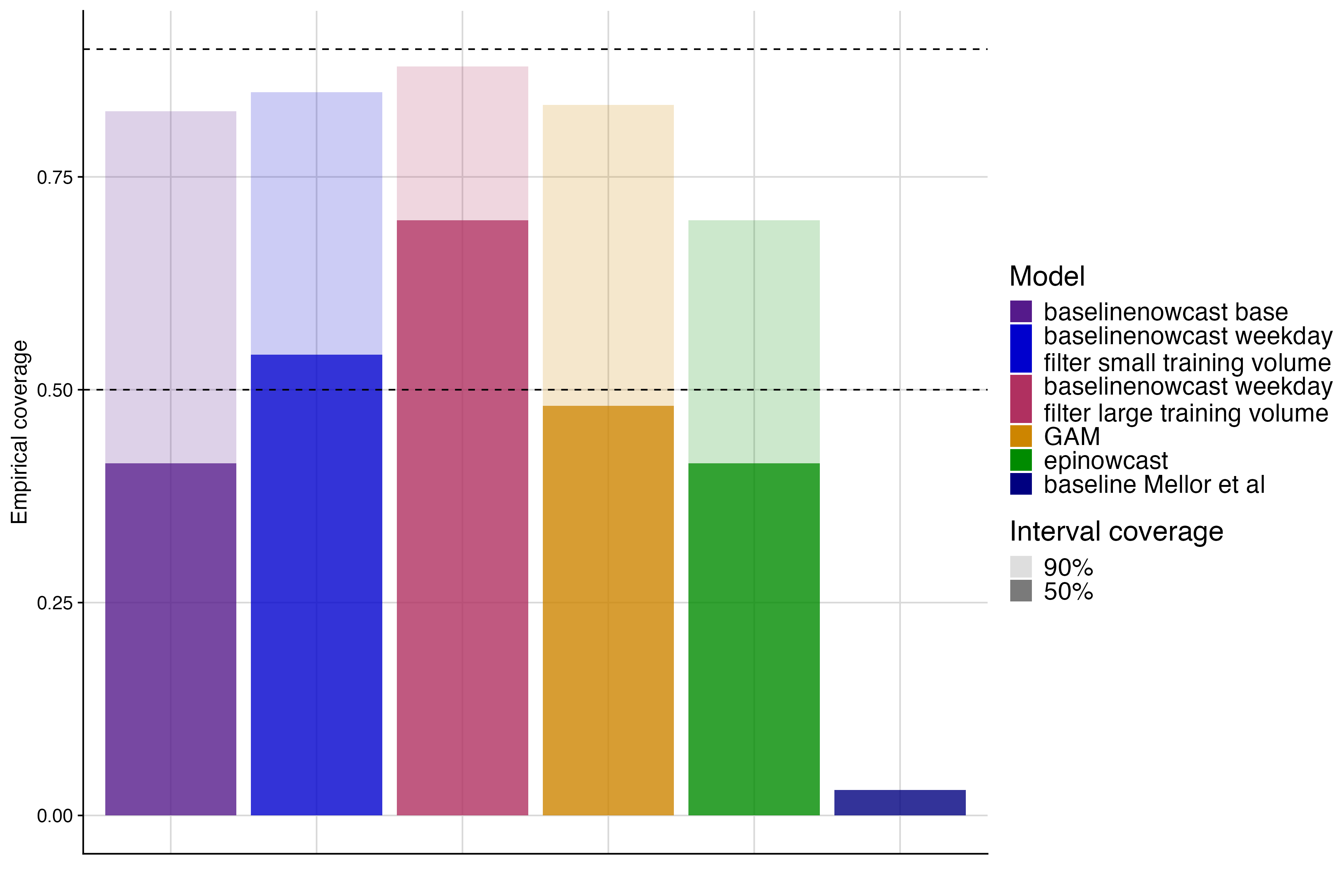


Fig S15. Empirical coverage at 50% and 90% prediction intervals for each model. colour indicates model, shading indicates prediction interval coverage, horizontal dashed lines indicate target prediction intervals.

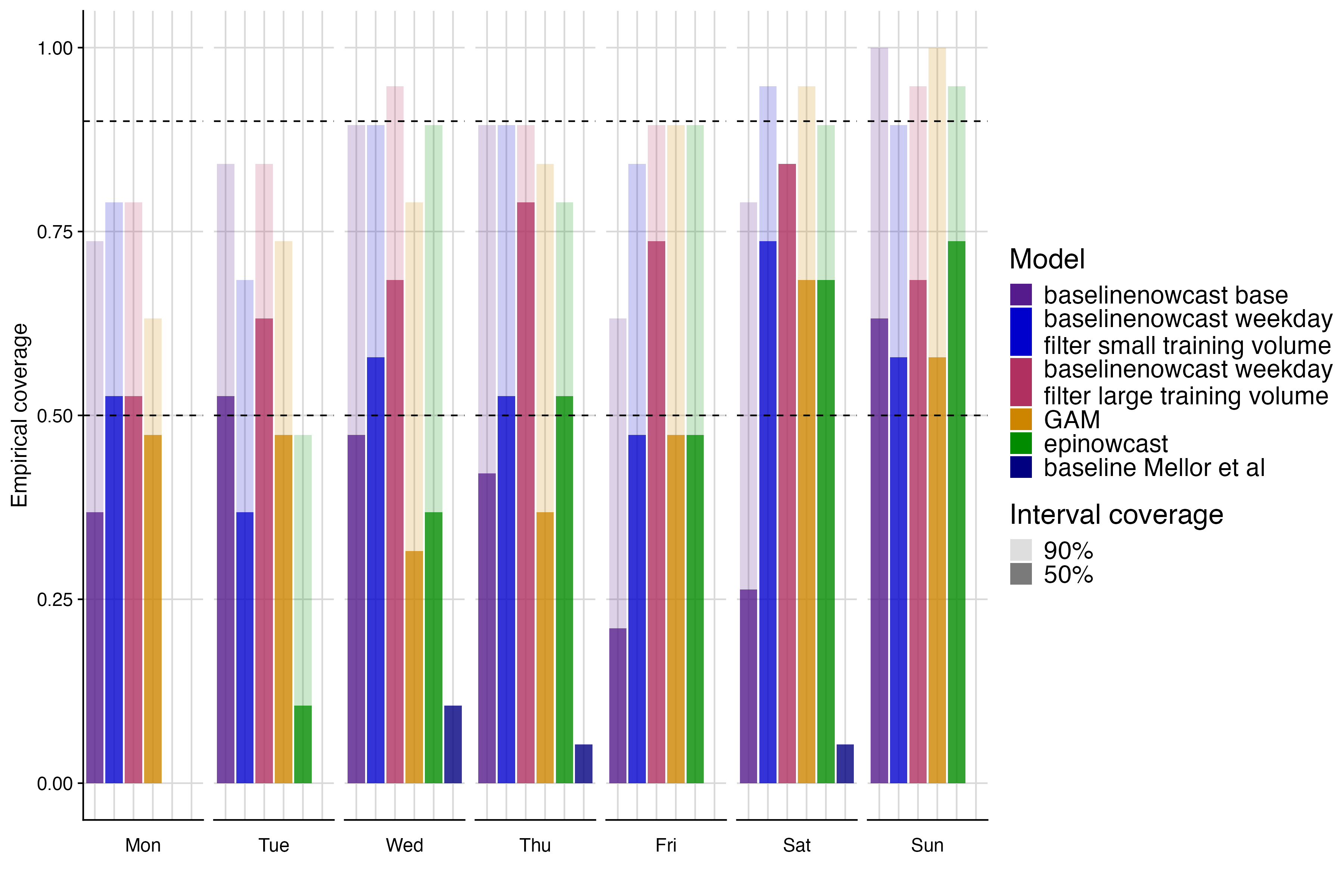


Fig S16. Empirical coverage at 50% and 90% prediction intervals by day of week for each model. colours indicate model, shading indicates prediction interval coverage, horizontal dashed lines indicate target prediction intervals.

## References

1. Wolffram D, Abbott S, Heiden M an der, Funk S, Günther F, Hailer D, et al. Collaborative nowcasting of COVID-19 hospitalization incidences in germany. PLOS Computational Biology [Internet]. 2023 Aug;19(8):1–25. Available from: <https://doi.org/10.1371/journal.pcbi.1011394>

2. Friedland J. Estimating unpaid claims using basic techniques. Casualty Actuarial Society. Casualty Actuarial Society, <https://www.casact.org/sites/default/files/database/studynotes_friedland_estimating.pdf>; 2010.

3. Inc. WR. Mathematica, Version 14.2 [Internet]. Available from: <https://www.wolfram.com/mathematica>

4. KIT Metrics Lab, Johannes Bracher, Melanie Schienle. RESPINOW-Hub: Nowcasting and Short-term Forecasting Platform for Respiratory Diseases [Internet]. Karlsruhe Institute of Technology; GitHub; 2025 [cited 2025 Jun 25]. Available from: <https://github.com/KITmetricslab/RESPINOW-Hub>

5. KIT Metrics Lab, Johannes Bracher, Daniel Wolffram, Melanie Schienle. Hospitalization Nowcast Hub: Collecting nowcasts of the 7-day hospitalization incidence in germany [Internet]. Karlsruhe Institute of Technology; GitHub; 2025 [cited 2025 Jun 25]. Available from: <https://github.com/KITmetricslab/hospitalization-nowcast-hub>