Combining Graph Neural Networks and Spatio-temporal Disease Models to Predict COVID-19 Cases in Germany

Cornelius Fritz[a,*], Emilio Dorigatti[a,b,*], David Rügamer[a,**]

[a] *Department of Statistics, Ludwig Maximilian Universität, München, Germany*
[b] *Helmholtz Zentrum München – German Research Center for Environmental Health, Neuherberg, Germany*

Abstract

During 2020, the infection rate of COVID-19 has been investigated by many scholars from different research fields. In this context, reliable and interpretable forecasts of disease incidents are a vital tool for policymakers to manage healthcare resources. Several experts have called for the necessity to account for human mobility to explain the spread of COVID-19. Existing approaches are often applying standard models of the respective research field. This habit, however, often comes along with certain restrictions. For instance, most statistical or epidemiological models cannot directly incorporate unstructured data sources, including relational data that may encode human mobility. In contrast, machine learning approaches may yield better predictions by exploiting these data structures, yet lack intuitive interpretability as they are often categorized as black-box models. We propose a trade-off between both research directions and present a multimodal learning approach that combines the advantages of statistical regression and machine learning models for predicting local COVID-19 cases in Germany. This novel approach enables the use of a richer collection of data types, including mobility flows and colocation probabilities, and yields the lowest MSE scores throughout our observational period in our benchmark study. The results corroborate the necessity of including mobility data and showcase the flexibility and interpretability of our approach.

Keywords: Deep Learning, Distributional Regression, Graph Neural Networks, Mobility Data, Uncertainty Quantification

1. Introduction

In December 2019, the region of Wuhan, China, experienced an outbreak of a novel coronavirus, initially infecting around 40 people [1]. Impeded by the early

*Equal contribution
**Corresponding author

findings that patients are already infectious in the pre-symptomatic stage of the disease [2] and that transmission occurs mostly either through the exchange of virus-containing droplets [3] or expiratory particles [4], i.e., aerosols, the containment strategies were not sufficient in detaining the virus from becoming a pandemic. Consequentially, the World Health Organization declared the virus a pandemic in March 2020 with 66.2 million infections and 1.5 million deaths worldwide as of December 7th, 2020 [5].

Given this development, it was repeatedly pondered how and if mathematical modeling can help contain the current COVID-19 crisis [6]. We argue that data science and machine learning can provide urgently needed tools to doctors and policymakers in various applications. For instance, model-assisted identification and localization of COVID-19 in chest X-rays can support doctors at achieving correct and precise diagnoses [7]. Meanwhile, policymakers benefit from studies determining and evaluating specific policies [8, 9]. In one noteworthy example, [10] were able to quantify to what extend targeted non-pharmaceutical interventions (NPIs) aided in ceasing the exponential growth rate of COVID-19. This work is often quoted as the main driver of the social distancing measures implemented in the UK, thus allowing the British government to pursue evidence-based containment strategies.

In order to adequately evaluate the role of specific policies and implement a successful containment strategy, a robust and interpretable forecast of the pandemic's state into the future is necessary. Among other purposes, this endeavor allows authorities to better manage healthcare resources (hospital beds, respirators, vaccines etc.).

The corresponding modeling task is broad and can be tackled at various levels of spatio-temporal granularity. While some proposals operate on daily country-level data [11, 12], others are designed to provide local predictions [13, 14]. From a methodological point of view, most of these approaches are influenced by either epidemiology, time-series analysis, regression models, machine or deep learning. However, we argue that the most promising approaches combine ideas from seemingly distant research areas with new types of data. Thereby, one can bypass restrictions of simpler models and improve the model's forecasting performance while also benefiting from the merits of each respective idea. In addition, allowing for the inclusion of novel data modalities in some of the more traditional approaches may further improve models. This was highlighted in previous literature by [15, 16], who suggest to include non-standard data sources, e.g., aggregated contact patterns obtained from mobile phones or behavioral data, into the analysis to help in understanding and fighting COVID-19.

Hybrid modeling approaches. Examples of these types of hybrid models are scattered throughout the literature. [13] combine a mechanistic metapopulation commonly used in epidemiology with clustering and data augmentation techniques from machine learning to improve their forecasting performance. For this endeavor, additional data sources, including news reports and internet search activity, were leveraged to inform the global epidemic and mobility model, an

epidemiological model already successfully applied to the spread of the Zika virus [17]. In another notable application, [9] enrich a relatively simple metapopulation model with mobility flows to numerous points of interest. Subsequently, this model is used to predict the effect of reopening after a specific type of lockdown through counterfactual analysis. With the help of Facebook, [18] utilize mobility and aggregated friendship networks to discover how these networks drive the infection rates on the local level of federal districts in Germany. In this context, another route to accommodate for such network data is employing a graph neural network (GNN). This technique builds on the intuitive idea of message passing [19] and has recently attracted a lot of attention in the deep learning community. For graph neural networks there is a wide range of possible applications, namely node classification or forecasting [20, 21]. In several occasions, this model class has already been applied to forecast the number of COVID-19 cases in 2020.

Applying GNNs to COVID-19 data. [22] construct a graph whose edges encode mobility data for a given time point collected from Facebook. Their approach exploits a long short-term memory (LSTM) architecture to aggregate latent district features obtained from several graph convolutional layers and transfer learning that accounts for the asynchrony of outbreaks across borders. In a comparable proposal, [23] employ a graph neural network to encode spatial neighborhoods and a recurrent neural network to aggregate information in the temporal domain. Through a novel loss function, they simultaneously penalize the squared error of the predicted infected and recovered cases as well as the long-term error governed by the transmission and recovery rates within traditional Susceptible-Infectious- Recovered (SIR) models. Contrasting these approaches, [24] propose a recurrent neuronal network to derive latent features for each location, hence they construct a graph whose edge weights are given by a self-attention layer. Instead of using a recurrent neural network, [25] construct a spatio-temporal graph by creating an augmented spatial graph that includes all observed instances of the observed network side-by-side and enables temporal dependencies by connecting each location with the corresponding node in previous days.

Our contribution. In this work, we propose a novel fusion approach that directly combines dyadic mobility and connectedness data derived from the online platform Facebook with structural and spatial information of Germany's cities and districts. In contrast to [18], we learn each district's embedding in the network in an end-to-end fashion, thus there is no need for a separate pre-processing step. With this, we heed recent calls such as [15, 16] highlighting the need for more flexible and hybrid approaches taking also dyadic sources of information into account. From a methodological point of view, we make this possible by combining graph neural networks with epidemiological models [26, 27] to simultaneously account for network-valued data and tabular data. We further provide comparisons, sanity checks as well as uncertainty quantification to investigate the reliability of our proposed model. In our application case, we provide fore-

casts of weekly COVID-19 cases with disease onset at the local level of 401 federal districts in Germany as provided by the Robert-Koch Institute.

This article's remainder is structured as follows. In the following section, the employed data are described in detail to determine several caveats that need to be accounted for when working with official surveillance data of an ongoing pandemic. Consecutively, Section 3 lays out the groundwork of distributional regression and graph neural networks that we combine in our main contribution, proposed in Section 4. To ascertain the practical use of this proposed model, Section 5 applies the proposed approach to Germany's weekly data. Finally, our article ends with a discussion and concluding remarks in Section 6.

2. Data

We distinguish our data sources between Facebook and infection data. While the infection data are time-series that are solely utilized in our model's structured and target component, most of the network data are directly used in the GNN module. To allow for sanity checks and interpretable coefficients, the networks are also transformed onto the units where we measured the time-series by calculating specific structural characteristics from the networks following [18].

2.1. Facebook data on human mobility and connectedness

To quantify the social and mobility patterns on the regional level, we use data on friendship ties, colocation probabilities, and district-specific data on the proportion of people staying put provided by Facebook [28]. These spatial data sets were made available through the *Data for Good* program and used in several other publications [29, 18, 30, 31, 32]. The spatial units are on the NUTS 3 level and encompass $n = 401$ federal districts. All data were collected from individual mobile phone location traces of Facebook users that opted in the *Location History* setting on the mobile Facebook application. For data security reasons, these individual traces were subject to differential privacy and aggregated to the 401 federal districts. As a result, the nodes of all networks are these districts. We augment the given network data with spatial networks encoding neighboring districts and distances in kilometers, that are respectively denoted by $x^N \in \{0,1\}^{n\times n}$ and $x^D \in \mathbb{R}^{n\times n}_{>0}$.

Colocation networks. The first type of network data that we incorporate in our forecast are colocation maps, which indicate the probability of two random persons from two districts to meet one another during a given week. We obtain for each week t a colocation matrix $x^C(t) \in \mathbb{R}^{n\times n}_{>0}$, where then ijth entry $(x^C_{ij}(t))$ indicates the probability of an arbitrary person from district i to meet another person from district j. To incorporate such network-valued data in the structured part of our framework, we transform it to tabular data. More specifically, we follow [18] and use the Gini index to measure the concentration of meeting

patterns of districts. For district $i \in \{1, ..., n\}$ this index can be defined through:

$$x_i^G(t) = \frac{\sum_{m,n \neq i} |x_{im}^C(t) - x_{in}^C(t)|}{2n \sum_{j \neq i} x_{ij}^C(t)} \in [0, 1].$$

Higher values translate to a restricted meeting pattern within a district, while lower values suggest rather diffused social behavior. In our application, we perform a weekly standardization of $x_i^G(t)$.

Social connectedness network. Secondly, we quantify social connections between the districts via Facebook friendships. Utilizing information from a snapshot of all Facebook connections within Germany of April 2020, we derive a Social Connectedness Index, that was first introduced by [33]. This pairwise time-constant index relates to the relative friendship strength between two districts and is stored in a weighted network $x^S \in \mathbb{R}_{>0}^{n \times n}$. For $i, j \in \{1, \ldots, n\}$ the entries of x^S are given by:

$$x_{ij}^S = \frac{\#\{\text{Friendship Ties between users in district } i \text{ and } j\}}{\#\{\text{Users in district } i\}\#\{\text{Users in district } j\}}.$$

Via multidimensional scaling, we can obtain two-dimensional district-specific embeddings from x^S, which we denote by x_i^S and incorporate in the structured component.

Staying put. Besides, we incorporate the weekly percentage of people staying put as a measure for people's compliance (Facebook users) with social distancing. We derive the corresponding weekly district-specific measure $x_i^{SP}(t)$ by averaging daily measures provided by Facebook. In this context, *staying put* is defined as being observed in one $0.6km \times 0.6km$ square throughout a day [34].

2.2. Time-series of daily COVID-19 infections

Current data on the pandemic's state in Germany are provided by the Robert-Koch-Institute. From this database, we obtain the number of people with symptom onset and registered cases of COVID-19 grouped by age, gender, and federal district (NUTS-3 level) for each day. Due to the observation that mainly people aged between 15 and 59 years are active users of Facebook, we limit our analysis to the corresponding age groups, namely, people with age between 15-35 and 36-59.

As discussed in [35], the central indicator of the infection occurrence is the number of people with disease onset at a specific day; hence our application focuses on the respective quantity. Due to mild courses of infection and inconsistent data collection, the data of disease onset is not known in about 30% of the cases. Therefore, we impute the missing values by learning a probabilistic model of the delay between disease onset and registration date. Eventually, we attain $y_{ig}(t)$, the infection counts of district i and group g during week t, by sampling from the estimated distribution of delay times. By doing that, we follow [18] where the procedure is given in more detail. The groups g are elements

of the Cartesian product of all possible age and gender groups. We denote all corresponding features by the vector $\boldsymbol{x}_{ig}(t)$.

Further, we are given data on the population sizes of each group g and district i from the German Federal Statistical Office, denoted by pop_{ig}, based on which we compute the population density den_{ig}. In this setting, one can assume that not the count but rate of infections in a specific district and group carries vital information. Hence, the target we model is the corresponding rate defined by $\tilde{y}_{ig}(t) = \frac{y_{ig}(t)}{pop_{ig}}$.

3. Methodological background

First, we present the methodological foundations of our approach: distributional regression and graph neural networks. To begin with, we will introduce distributional regression approaches on the basis of which we define our deep probabilistic model in Section 4. To incorporate modeling techniques from statistics and epidemiology in our network, we use so-called structured additive predictors that represent smooth additive effects of input features and can be represented in a neural network. As we will elaborate in our main section, effect smoothness can be achieved by a specifically designed network regularization term. The second building block of our model are graph neural networks, which we introduce subsequently.

3.1. Distributional regression

Distributional regression is a modeling approach to regress a (parametric) distribution $\mathcal{D}$ on p given input features $\boldsymbol{x} \in \mathbb{R}^p$ [36]. In contrast to other regression approaches that, e.g., do only relate the mean of an outcome variable to certain features, distributional regression also accounts for the uncertainty of the data distribution, known as aleatoric uncertainty [37]. Given a parametric distributional assumption $\mathcal{D}(\theta_1, \ldots, \theta_K)$, the model learns the distributional parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^\top$ by means of feature effects. In structured additive distributional regression [38], each distributional parameter is estimated using an additive predictor $\eta_k(\boldsymbol{x}_k)$. In this context, $\eta_k : \mathbb{R}^{p_k} \to \mathbb{R}$ is an additive transformation of a pre-specified set of features $\boldsymbol{x}_k \in \mathbb{R}^{p_k}$, $1 < p_k < p$. This additive predictor is finally transformed to match the domain of the respective parameter by a monotonic and differentiable transformation function h_k:

$$\theta_k(\boldsymbol{x}_k) = h_k\big(\eta_k(\boldsymbol{x}_k)\big).$$

Note that the K parameters relating to $\mathcal{D}$ now depend on the features.

Moreover, structured additive distributional regression models allow for a great variety of feature effects in the additive predictor that are interpretable by nature [39]. Examples include linear, non-linear, or random effects of one or more features. The latter two effect types can be represented via basis functions (such as regression splines, polynomial bases or B-splines). Further examples and details can be found, e.g., in [40, 39]. Due to the additivity of effects in η_k, the influence of single features or a combination of features can in many cases be directly related to the mean or the variance of the modeled distribution.

Semi-structured deep distributional regression. A recent trend is the combination of neural networks with statistical regression models in various ways [41, 42, 43, 44, 45]. In this work, we make use of *semi-structured deep distributional regression* [SDDR, 46]. SDDR combines neural networks and structured additive distributional regression by embedding the statistical regression model in the neural network and ensures the identifiability of the regression model part. SDDR is a natural extension of distributional regression by extending the additive predictor η_k of each distributional parameter with its *structured effects* to latent features, so-called *unstructured effects*, that are learned by one or more deep neural network (DNN). To disentangle the structured model parts from the unstructured parts, an orthogonalization cell is used to ensure identifiability of the structured model effects.

3.2. Graph neural networks

Graphs are a mathematical description of the entities and their relationships in a given domain and naturally arise in a variety of seemingly distant fields. However, due to their non-euclidean structure, it is not straightforward to apply traditional machine learning methods to problems involving graphs, as these methods operate on vectors [21]. A number of methods have been proposed to embed graphs into low-dimensional euclidean spaces, allowing to use the resulting vector representations for prediction tasks with traditional machine learning algorithms such as node classification and missing link prediction [20]. Inspired by the success of convolutional neural networks, comparable approaches formulated convolutions for graphs via the spectral domain relying on the convolution theorem [47]. Later versions of convolutional operators were adapted to the vertex domain by means of a message-passing framework [19]. In this framework the feature vector of each node is updated based on the feature vectors of its own neighbors and the edges connecting them [48]. Following this, more advanced neighborhood aggregation methods [49], scalable inference [50] and domain-specific applications [19] have been proposed. In general, a graph neural network performs R rounds of message passing, after which all nodes' latent features can be combined to obtain a unified representation for the whole graph [51], or individual nodes [52]. We can use the latter type of representations to derive node-specific predictions.

Following the notation in [19], in each message-passing round r the neighbors $N(v)$ of v are aggregated into a message $\boldsymbol{m}_v^r$ through a message function M_r:

$$\boldsymbol{m}_v^r = \sum_{w \in N(v)} M_r(\boldsymbol{x}_v^r, \boldsymbol{x}_w^r, \boldsymbol{e}_{vw}) \tag{1}$$

where $\boldsymbol{x}_v^r$ denotes the latent features of node v after r rounds of message passing and $\boldsymbol{e}_{vw}$ the features associated to the edge connecting v and w. Next, an update function U_r combines $\boldsymbol{x}_v^r$ and $\boldsymbol{m}_v^r$ to obtain the updated latent features $\boldsymbol{x}_v^{r+1}$:

$$\boldsymbol{x}_v^{r+1} = U_r(\boldsymbol{x}_v^r, \boldsymbol{m}_v^r) \tag{2}$$

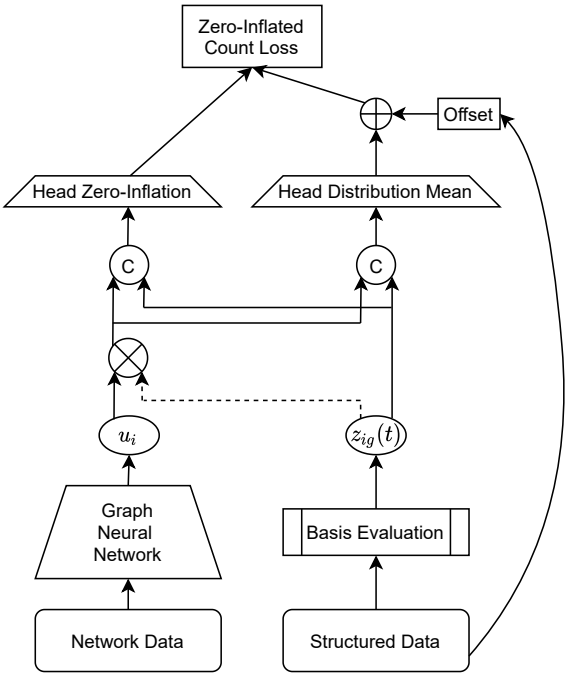

Figure 1: Network architecture of our proposed model. The mobility data is fed into a GNN on the bottom left to learn latent features $\boldsymbol{u}_i$. On the bottom right side, the structured data is transformed using basis evaluation. Using the orthogonalization the learned effects $\boldsymbol{u}_i$ are projected onto the orthogonal complement of selected parts of the basis evaluated structured features. Both parts are combined using a concatenation and fed into both a network head that learns the zero-inflation as well as a network head to learn the distribution's mean. After adding an offset to the mean, both parts are finally combined in a distributional layer that learns the zero-inflated count distribution based on the corresponding log-likelihood.

Taken together (1) and (2) define a message passing round that propagates information one hop further than the previous round. Multiple message passing rounds can be applied successively to diffuse information across the complete network.

4. Combining network-valued and spatio-temporal disease data

We present our hybrid modeling approach to forecast weekly district-wise COVID-19 cases based on structured and unstructured data sources described in Section 2. The general framework is depicted in Figure 1 and fuses the interpretability of distributional regression with a GNN architecture to flexibly learn all district's latent representation from the network data.

4.1. Neural network formulation

Distributional assumption. Our considered time window stretches over a low-infection phase during which 20 - 30% of the observations reported no cases.

This phase is delimited by two primary infection waves, approximately from March - April and October - November, showing more than 1,200 cases per week. Hence, our goal is to build a model that can adequately predict high numbers as well as zero observations, which are common during low-infection seasons. We achieve this by assuming that the cases follow a mixture distribution of a point mass distribution at zero and an arbitrary count distribution with mixture weight $\pi \in [0, 1]$. Here, we regard any probability distribution defined over non-negative integers as a count distribution, e.g., the negative binomial, Poisson, and generalized Hermite distribution [53]. The resulting *zero-inflated* distribution $\mathcal{D}$ is primarily characterized by the mean of the count component λ and the zero-inflation probability π. Any additional parameters relating to other traits of the count distribution, e.g., the scale parameter of the negative binomial distribution, are denoted by χ. The probability mass function of $\mathcal{D}$ is given by:

$$f_{\mathcal{D}}(y|\lambda, \pi, \chi) = \pi I(y = 0) + (1 - \pi) f_{\mathcal{C}}(y|\lambda, \chi), \tag{3}$$

where $f_{\mathcal{C}}$ is the density of the chosen count distribution $\mathcal{C}$ and I denotes the indicator function. By incorporating the point mass distribution, the model can capture excess rates of zero observations and π may be interpreted as the percentage of excess-zero observation [54].

In our modeling approach for COVID-19 cases $y_{ig}(t)$ we relate structured data as well as network data to the parameters λ and π of the zero-inflated distribution, which yields the following distributional and structural assumption:

$$\begin{aligned} y_{ig}(t) &\sim \mathcal{D}(\lambda_{ig}(t), \pi_{ig}(t), \chi), \\ \lambda_{ig}(t) = h_1\left(\eta_{1,ig}(t)\right),\ \pi_{ig}(t) &= h_2(\eta_{2,ig}(t)), \end{aligned} \tag{4}$$

with chosen zero-inflated distribution $\mathcal{D}$. For the structural component in (4), the two feature-dependent distributional parameters are described through the additive predictors $\eta_{1,ig}(t)$ and $\eta_{2,ig}(t)$. We transform these predictors via fixed transformation functions h_1, h_2 to guarantee correct domains of the respective modeled parameter, e.g., a sigmoid function for the probability $\pi_{ig}(t)$ (see Section 3.1).

Additive predictors. Inspired by SDDR [46], we estimate additive effects of tabular features on the parameters characterizing the zero-inflated distribution using a single-neuron output layer. As proposed in various statistical COVID-19 modeling approaches [18, 55], we learn these structured effects with an appropriate regularization to enforce smoothness of non-linear effects. This penalization can be seen as a trade-off between complexity and interpretability [56].

The additive predictors $\eta_{1,ig}(t)$ and $\eta_{2,ig}(t)$ can be defined in terms of both unstructured and structured features (left and right bottom input in Figure 1). In the structured model part, we use the complete suite of district-specific features, defined in Section 2, including them as arbitrary additive effects, which are detailed in the next section. In the following, we make this clear by using $\boldsymbol{z}_{ig}(t)$, which are the input features $\boldsymbol{x}_{ig}(t)$ but transformed using some basis function evaluation. We denote the corresponding feature weights by

$\boldsymbol{\vartheta}^{\text{str}} = (\boldsymbol{\vartheta}_1^{\text{str}\top}, \boldsymbol{\vartheta}_2^{\text{str}\top})^\top$ corresponding to η_1 and η_2, respectively. The unstructured part of our network computes each district's embedding (node) by exploiting time-constant district population attributes and edge attributes. For our application, these attributes encode geographic connectedness between districts and social connectedness. Here, the message passing framework enables the embeddings to contain first, second, and higher-order information about the spread of the disease among all districts. Finally, we inform either additive predictors with the embeddings $\boldsymbol{u}_i$ for district i learned from the GNN to incorporate the network data in the distributional framework.

Orthogonalization. Identifiability is crucial to our analysis since some feature information is shared between the structured effects and unstructured effects. For instance, the social connectedness index x^S is exploited in the structured part via the MDS-embeddings but also in the graph neural network as an edge attribute. If these two model parts are not adequately disentangled, it is unclear what part of the model is accounting for which information in the shared features. Therefore, the latent GNN representations $\boldsymbol{u}_i$ are orthogonalized with respect to $\boldsymbol{z}_{ig}(t)$ yielding $\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i \otimes \boldsymbol{z}_{ig}(t)$. More specifically, we project $\boldsymbol{u}_i$ in the orthogonal complement of the column space spanned by $\boldsymbol{z}_{ig}(t)$ and use $\tilde{\boldsymbol{u}}_i$ instead of $\boldsymbol{u}_i$ in the additive predictors (see [46] for further details). In the final step, we combine the structured and unstructured effects as a sum of linear orthogonalized embedding effects and the structured predictors, i.e., $\eta_{k,ig} = \boldsymbol{x}_{ig}(t)\boldsymbol{\vartheta}_k^{\text{str}} + \tilde{\boldsymbol{u}}_k\boldsymbol{\vartheta}_k^{\text{unstr}}$, $k \in \{1, 2\}$.

Different exposures in count regression. As already discussed in Section 2, the primary goal of our application is modeling the rates of infections rather than the raw observed counts, as each state is subject to a different exposure of COVID-19. Still we learn the mean of the counting distribution in (4) and correct for the differing population sizes by adding a constant offset term to the concatenated linear predictor, i.e., we add $\log(pop_{ig})$ to $\eta_{1,ig}(t)$. For additional information on this procedure in the realm of zero inflated models, we refer to [57].

4.2. Proposed COVID-19 model specification

Distributional regression. On the basis of (4), we set up our COVID-19 model as follows: let $\mathcal{D}$ define a zero-inflated Poisson (ZIP) distribution. We reparameterize the distribution in terms of only one additive predictor $\eta_{ig}(t) = \eta_{1,ig}(t)$ as this proved to help numerical stability. We therefore define $\pi_{ig}(t) \equiv \exp(-\exp(\chi + \log(\lambda_{ig}(t))))$ and learn the distribution's rate via $\lambda_{ig}(t) = \exp(\eta_{ig}(t))$.

Another option for modeling counts is to use a negative binomial (NB) distribution as, e.g., done by [18]. The NB distribution is often chosen over the Poisson distribution due to its greater flexibility, particularly by allowing to account for overdispersion. We will compare the NB distribution against the ZIP approach by reparameterizing the NB distribution in terms of its mean, similarly to what did for the ZIP. Table 1 further summarizes the use of all available features in our model using their transformation and incorporation in the structured additive as well as GNN model part.

Feature	Structured Part Basis Evaluation (Transformation)	GNN Node	 Edge
$\tilde{y}_{ig}(t-1)$	TP-Spline (logp1$(\cdot)$)		
$x_i^G(t)$	Bivariate Spline with t		
x^S	-		✓
x_i^S	Bivariate Spline		
x^D	-		✓
x^N	-		✓
$x_i^{SP}(t)$	Bivariate Spline with t		
pop_{ig}	Offset (log$(\cdot)$)	✓	
den_{ig}	-	✓	
g	Dummy Variables		

Table 1: Features and their incorporation into the structured and GNN part of our model. For each feature, the second column indicates the basis function evaluation used in the structured part, which is applied to the the feature itself or a transformation of it given in brackets. If no transformation is given, the identity is used. For bivariate effects we use bivariate thin plate (TP) regression splines. The transformation $\text{logp1}(y) = \log(y+1)$. The third column indicates the incorporation of each feature in the GNN part, either as a node or edge feature. To also account for the group-specific nature of each distributional parameter, we further add a gender and age effect using g as a dummy variable.

Graph neural network. Among all possible variants of the general message passing framework described above, we opt for the proposition of [58], thereby making use of edge attributes and efficiently handling our relatively large graphs. The message function of (1) thus becomes:

$$M_r(\boldsymbol{x}_v^r, \boldsymbol{x}_w^r, \boldsymbol{e}_{vw}) = \frac{1}{H^r \cdot |N(v)|} \sum_{h=1}^{H^r} w_h^r(\boldsymbol{e}_{vw}) \cdot \boldsymbol{\Theta}_h^r \boldsymbol{x}_w^r \tag{5}$$

which uses H^r linear maps to transform the neighbors' features and H^r radial basis function (RBF) kernels to weight the linear maps:

$$w_h^r(\boldsymbol{e}) = \exp\left\{-\frac{1}{2}(\boldsymbol{e} - \boldsymbol{\mu}_h^r)^\top \text{diag}(\boldsymbol{\sigma}_h^{r\,2})^{-1}(\boldsymbol{e} - \boldsymbol{\mu}_h^r)\right\}. \tag{6}$$

Whereas the node update function of (2) becomes:

$$U_r(\boldsymbol{x}_v^r, \boldsymbol{m}_v^r) = \boldsymbol{m}_v^r + \boldsymbol{\Theta}_0 \boldsymbol{x}_v^r + \boldsymbol{b}^r \tag{7}$$

with trainable parameters $\boldsymbol{\Theta}_0^r$, $\boldsymbol{\Theta}_1^r, \ldots, \boldsymbol{\Theta}_{H^r}^r$, $\boldsymbol{\mu}_h^r$, $\boldsymbol{\sigma}_h^r$ and $\boldsymbol{b}^r$.

Because people can travel from any district to any other district, e.g., via train or car, we use a fully connected graph as input to the GNN and embedded information about social connectedness in the edge attributes. The first graph convolutional layer uses $H^1 = 8$ affine maps with output dimensionality of 256, followed by a the second layer that further reduces this number to 128 latent components with $H^2 = 4$. Next, four fully-connected layers successively reduce the dimensionality of the node embeddings to 16 components. All layers use

batch normalization [59] followed by leaky ReLU activation [60]. To reduce the chance of overfitting, we further use dropout [61] with probability 0.25 after the two graph convolutions.

4.3. Uncertainty quantification

A crucial tool to investigate the model's reliability is to assess its uncertainty. While the proposed approach explicitly models the uncertainty in the given data distribution (aleatoric uncertainty), we can also derive epistemic uncertainty of parts of the model through its connection to statistical models.

Epistemic uncertainty. Using standard regression model theory, we can derive the epistemic uncertainty of our model, i.e., the uncertainty of model's weights. When regarding the GNN part of the model as a fixed offset $\boldsymbol{o}$ and fixing the amount of smoothness defined by $\boldsymbol{\xi}$, it follows

$$\boldsymbol{\vartheta}^{\text{str}} \mid \boldsymbol{y}, \boldsymbol{\xi}, \boldsymbol{o} \sim \mathcal{N}(\hat{\boldsymbol{\vartheta}}, (\hat{\boldsymbol{\mathcal{I}}}^{\text{str}} + \boldsymbol{P})^{-1}), \tag{8}$$

where $\hat{\boldsymbol{\mathcal{I}}}^{\text{str}}$ is the Hessian of the negative log-likelihood at the estimated network parameters $\hat{\boldsymbol{\vartheta}}$ [39] . We note that especially the conditioning on $\boldsymbol{o}$ (the GNN) neglects some additional variance in the parameter estimates but still allows us to get a feeling for the network's uncertainty.

To additionally account for the epistemic uncertainty of the GNN, we turn to deep ensembles [62], a simple method known to result in reliable uncertainty estimates. In this context, the epistemic uncertainty can be estimated simply by computing the standard error of the predictions of an ensemble of models, each trained from scratch with a different random initialization.

Aleatoric uncertainty. In addition to epistemic uncertainty, our model accounts for aleatoric uncertainty by modeling all the distributional parameters of the zero-inflated count distribution explicitly. For example, in the case of the ZIP distribution, the distribution's variance can be derived from its parameters as follows:

$$(1 - \pi_{ig}(t)) \cdot (\lambda_{ig}^2(t) + \lambda_{ig}(t)\chi) - ((1 - \pi_{ig}(t)) \cdot \lambda_{ig}(t)^2. \tag{9}$$

In particular, this allows us to make a probabilistic forecast for all weeks, districts, and each cohort (age, gender). After having observed the forecasted values, we can assess how well the model performed and how well it predicted uncertainty of the data distribution when only trained with historic data up to a certain week.

4.4. Network training

We train model (4) by minimizing the negative log-likelihood derived from the distributional assumption in (4). The combined $\rho := p_1 + p_2 + \tau + p_\chi$ weights $\boldsymbol{\vartheta} \in \mathbb{R}^\rho$ of the whole network subsume p_1 weights for the first additive predictor η_1, p_2 effects for the second additive predictor η_2, τ effects for the GNN and

p_χ weights for additional distributional parameters. For readability, we omit in the following the indices i and g as well as the time-dependency of the two distributional parameters, yet make the dependency on learned weights explicit. Stemming from (3), we can construct a joint likelihood $\ell(\boldsymbol{\vartheta})$ by summing up the contribution of each individual observation, that is given by

$$\ell(\boldsymbol{\vartheta}) = \log\left(\pi(\boldsymbol{\vartheta})I(y=0) + (1-\pi)f_{\mathcal{C}}(y|\lambda(\boldsymbol{\vartheta}),\chi)\right). \tag{10}$$

Under conditional independence, feature weights can be learned by minimizing the sum of negative log-likelihood contributions for all observations. To avoid overfitting and estimate smooth additive effects in the structured part of our model, we add a quadratic penalty term $J(\boldsymbol{\vartheta}) = \sum_{j=1}^{2} {\boldsymbol{\vartheta}_j^{\text{str}}}^\top \boldsymbol{P}_j \boldsymbol{\vartheta}_j^{\text{str}}$. Thereby, we regularize weights in the network $\boldsymbol{\vartheta}_j \in \mathbb{R}^{p_j}$ that correspond to smooth structured effects. Penalization is controlled by individual complexity parameters that we incorporate in the penalty matrices $\boldsymbol{P}_j \in \mathbb{R}^{p_j \times p_j}, j = 1, \ldots, 2$. These matrices, in turn, are block-diagonal matrices that are derived by the chosen basis functions in the structured model part [39]. We do not additionally penalize the count parameter χ nor the GNN model part other than using the orthogonalization. In practice, we observe that training the network can be stabilized when choosing RMSprop [63] as the optimizer.

5. Application and evaluation

We now apply our model from Section 4.2 to the data introduced in Section 2. To this end, we first explain our evaluation scheme used to compare our model in a benchmark study against other alternative modeling approaches we describe in Section 5.2. In Section 5.3 we report the results of these model comparisons. We finally investigate various aspects of our model to further demonstrate how our approach allows for an intuitive interpretation and quantification of uncertainty.

5.1. Evaluation scheme

Apart from our primary goal to provide one-week forecasts, we also investigate our approach's behavior throughout the pandemic. Therefore, we apply an expanding window approach, where we use a certain amount of data from past weeks, validate the model on the current week, and forecast the upcoming week. We do this for different time points, starting with training until week 30, validation on week 31, and testing on week 32. In 3-week-steps, we expand the time window while adapting the validation and test week. In Figure 2 we sketch a visual representation of this scheme.

5.2. Model comparisons

We compare our approach against four other algorithms and different model specifications of our framework. As a baseline model, we use the mean of a sliding window approach applied to the given training data set (MEAN). The prediction on the test set then corresponds to the mean of the last week in

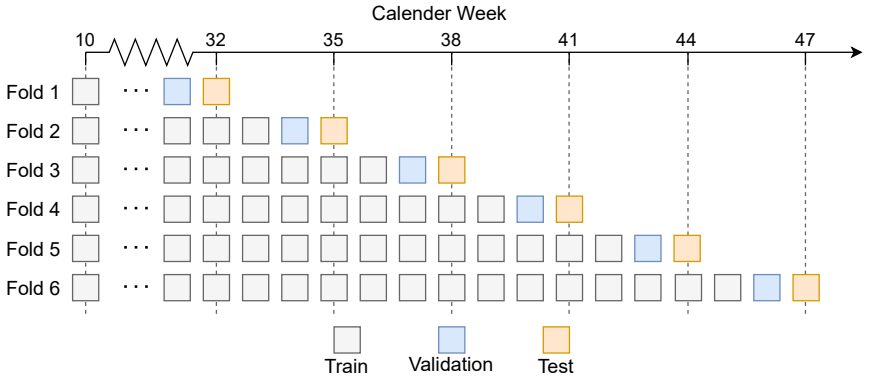

Figure 2: Evaluation Scheme based on an expanding window approach over the available historical weekly data.

the training data for each of the four subgroups (age, gender). Our statistical regression baseline is a generalized additive model (GAM) inspired by the work from [18], modeling the mean of a negative binomial distribution using various smooth predictors and tensor-product splines. We further apply gradient boosting trees as a state-of-the-art machine learning baseline. Due to its computational efficiency, scalability and predictive performance, we choose the well-known XGBoost implementation [64] in our comparisons. Finally, we compare our model against a vanilla deep neural network (DNN), a multi-layer perceptron with a maximum amount of 4 hidden layers with ReLU or tanh activation function, dropout layers in between and a final single output unit with linear activation. To enable a meaningful comparison, we corrected all benchmark model outputs for the differing exposures by incorporating an offset in same way as explained in Section 4.1. Similar to classical statistical models, this allows the model to learn the actual rate of infections. In all cases, we optimize the model using the Poisson log-likelihood (count loss). We furthermore tune the DNN and XGBoost model using Bayesian optimization (BO; [65]) with 300 initial sampled values for the set of tuning parameters and ten further epochs, each with 30 sampled values. Finally, we investigate the performance of a simple GNN, i.e., not in combination with distributional additive regression, optimized on the RMSE.

5.3. Results

First, we report forecasting performances on all data folds (based on the optimal setting found by BO). Consecutively, we scrutinize our model's behavior by examining partial effect plots and estimated uncertainties.

Forecasting performance. In Table 2 we give the forecast performances of our model and approaches described in Section 5.2. Out of the benchmark models, the GAM is the best performing models returning consistently smaller RMSE

	Calendar Week					
	32	35	38	41	44	47
XGBoost	4.926	5.188	7.447	**15.327**	65.036	74.235
DNN	10.179	12.178	17.897	64.065	108.474	80.901
GAM	4.042	4.738	4.736	21.666	18.556	23.813
MEAN	5.038	3.666	6.196	30.910	20.090	23.159
GNN	5.972	6.785	11.355	49.064	77.162	53.489
Ours (ZIP)	**3.931**	4.235	**4.500**	16.588	**17.738**	**15.050**
Ours (NB)	4.096	**4.094**	5.174	28.580	18.098	31.724

Table 2: RMSE values for different methods (rows) and folds (columns) corresponding to the data splitting approach explained in Section 5.1 and Figure 2. Bold numbers denote the best result in each fold across all models.

values than XGBoost and the DNN, with one exception in week 41. The rolling mean (MEAN) performs similar well before the second wave in Germany (Week 32 and 35). However, the numbers stagnate during Germany's second lockdown (Week 47), which may be seen as an external shock that cannot be accounted for by the previous weeks' rising numbers. The vanilla DNN yields the worst performance, where the BO found the smallest architecture with only one hidden layer with one unit to be the best option. While this result aligns with the good performance of MEAN and GAM, dropout in the DNN between the input and hidden layers does apparently not yield enough or the appropriate regularization to prevent the DNN from overfitting. Our model shows similar performance to the GAM model, which is again in line with our expectations, as we orthogonalize the GNN part of our network w.r.t. the whole structured additive part. In particular, the model performs notably better for the weeks 41 - 47 than the GAM, and yields the best or second best results compared to all other models on each fold. Although XGBoost outperforms our approach on week 41, its worse performance on all other folds does not make it a reasonable choice. The same holds for the NB variation of our approach, which delivers the second-best performances. Finally, the GNN itself yields reasonable predictions in the first two weeks but does not perform well for the other weeks.

Model interpretation. We now provide further insights into our model by analyzing the partial effects of its structured additive part. More specifically, we focus on two instances of our models that are trained with data until calendar week 30 and 44. We pick the respective weeks to showcase the partial effects during a high and low season of the pandemic.

To begin, we investigate the partial effects of the Gini coefficient (G) derived from the colocation maps and the percentage of people staying put (SP) on the left and right side of Figure 3, respectively. Moreover, a high standardized Gini coefficient translates to meeting behavior that is more dispersed than the average of all districts. Hence, a low standardized Gini coefficient (less mobility than average) leads to lower infection rates, especially between calendar week 10 and 30. For the percentage of people staying put, the temporal dynamics are

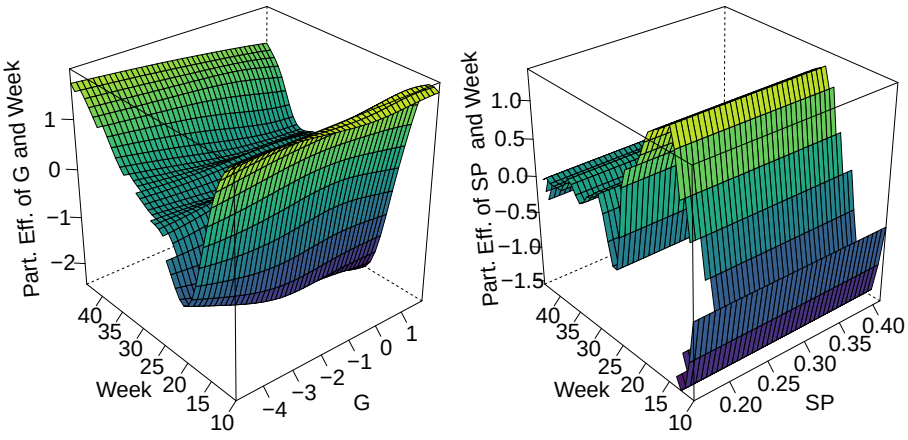

Figure 3: Estimated bivariate partial effects of the week and Gini coefficient (G) on the left as well as of the week and Percentage Staying Put (SP) on the right.

somewhat opposite and exhibit small effects in the first weeks and after week 30. Thereby, we may conclude that having a higher percentage of people staying put also lowers the infection rates. Further, we observe that the incorporated penalty term successfully regularizes the bivariate effect term to be a linear effect in the direction of the percentage of people staying put.

Epistemic uncertainty. As explained in Section 4.3, while an epistemic uncertainty for the structured part of our model can be derived theoretically, for the GNN part, we estimate our models' uncertainty through an ensemble. Specifically, we repeat the training procedure ten times, each time with a different random initialization. Consecutively, we compute the standard errors of all predictions, thus defining the epistemic uncertainty.

The epistemic uncertainty is well correlated (Spearman's $\rho = 0.76$) with the absolute error resulting from the mean prediction (Figure 4 left) and grows approximately linearly with it. However, the variability of the average error is not reliable in the last bin due to the low number of samples with high (15) epistemic uncertainty. The epistemic uncertainty is generally higher for high-incidence weeks such as week 44 compared to a low-incidence week such as week 30 (Figure 4 right). In addition, the ensemble has a very slight tendency to underestimate the number of cases for week 44 by 1.26, and to overestimate the cases for week 30 by 0.25. Although statistically significant (one-sided t-test, $t = 3.24$, $p = 0.001$ and $t = 3.38$, $p = 0.0007$, respectively), the resulting differences are practically irrelevant, hence suggesting that the ensemble is approximately calibrated. In general, this correlation between epistemic uncertainty and forecast error provides a reliable diagnostic of the trustworthiness of

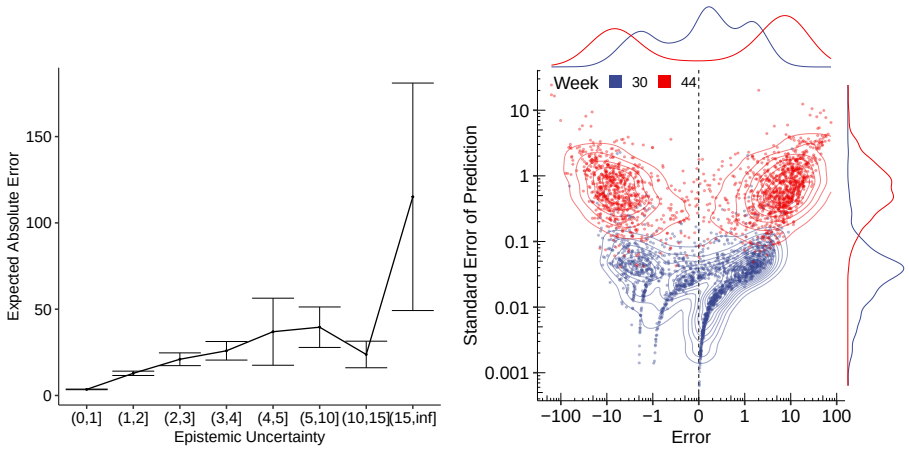

Figure 4: Left: average absolute error incurred by the ensemble for increasing levels of epistemic uncertainty, with vertical bars denoting ± one standard error of the estimation. Right: Standard error of the predictions of an ensemble of ten networks correlated with the error incurred when using the ensemble's mean prediction for each district, age, and gender cohort during a low-infection phase (week 30) and the second wave of infections (week 44).

our proposed model's predictions.

The partial effect of lagged infection rates in Figure 5 additionally depicts its epistemic uncertainty when the GNN weights are fixed. The figure's narrow shaded areas translate to high certainty of the partial effect from the respective feature. Moreover, the partial effect translates to the finding that the higher the infection rate was in the previous week, the higher the predictions are in the upcoming week. In line with this result prior studies, already identified this feature as a principal driver of the structured model part of our network [18].

Aleatoric uncertainty. We evaluate our ZIP model's aleatoric uncertainty by applying the expanding window training scheme analogous to our model evaluation. For each prediction, we calculate predictive distribution intervals using the mean prediction ± 2 times the standard deviation derived from (9). Figure 6 depicts the probabilistic forecasts of the modeled ZIP distribution for different districts in Germany. These districts constitute particularly difficult examples from relatively rural areas and cases in larger cities (Munich / München) as well as sites that were hardly affected and severely affected by the pandemic. In Figure 6 we visualize the true values as points and the predicted mean as a black line. Here, the shaded purple area symbolizes the predictive distribution intervals. We observe that most of the points are well within the given prediction interval, thus the distribution captures the dispersion in the data quite well. As expected from the Poisson distribution, results indicate that the aleatoric uncertainty increases with the rising number of infections. However, some intervals are not able to cover larger fluctuations and steeper increases of infections, such as is the case in *München*, *Gütersloh* or *Vorpommern-Rügen*.

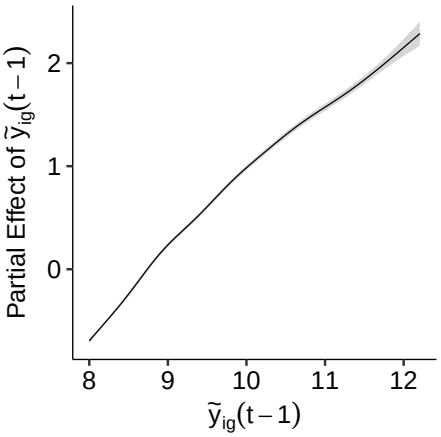

Figure 5: Estimated effect of lagged infection rate with corresponding confidence intervals for week week 44.

Overall, the intervals derived from the predictive distributions cover on average over 80% of all cases in all groups, weeks and districts. This indicates that the estimation of distribution variances for groups and weeks works well, but shows also room for improvement for later weeks where the distribution is not perfectly calibrated.

6. Discussion

Following several experts' call to account for human mobility in existing statistical and epidemiological models of COVID-19, we propose a multimodal network that fuses existing efforts with graph neural networks. We thereby enable the use of a more nuanced collection of data types, including mobility flows and colocation probabilities, in the forecasting setting. Our results indicate a notable improvement over existing approaches, which we achieved by accounting for the network data. The given findings also highlight the need for regularization and showcase how common ML approaches can not adequately capture the autoregressive term, which, in turn, proved to be essential for the forecast. Our model's investigation further showed that uncertainty can be well captured by the model, although further calibration may be vital for its aleatoric uncertainty.

Caveat. We want to emphasize that despite the convincing results, the given analysis only addresses a small subset of processes involved in the spread of COVID-19 and should not be the sole substance for decision making processes in the future. In particular, forecasting infection rates in the short run does not (need to) address reporting or observation biases typically present for such data and requires a solid data basis, which in the case of Germany, is provided by the Robert-Koch Institute.

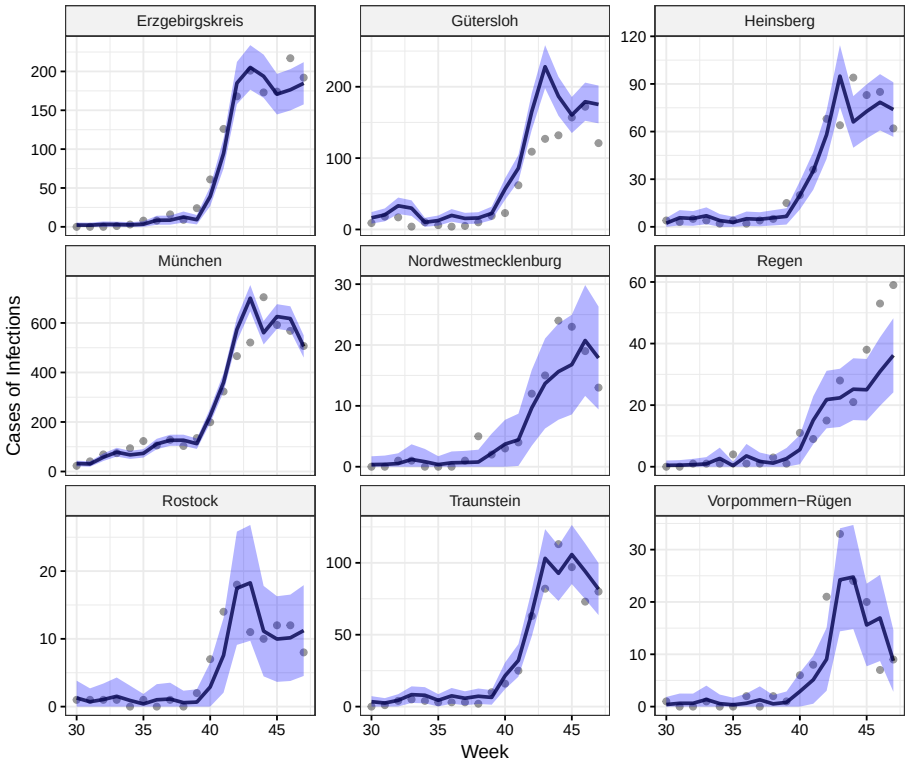

Figure 6: Exemplary prediction intervals for age group 35 - 59 and gender male and selected districts (facets) in Germany. For the different forecast weeks (x-axis) the true number of infections are visualized by points and contrasted with the model's prediction (black line) and the prediction interval (shaded purple area).

Acknowledgements

The work has been partially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. ED is supported by the Helmholtz Association under the joint research school "Munich School for Data Science - MUDS" (Award Number HIDSS-0006). The authors of this work take full responsibilities for its content.

References

[1] Y.-C. Wu, C.-S. Chen, Y.-J. Chan, The outbreak of COVID-19, Journal of the Chinese Medical Association 83 (2020) 217–220.

[2] C. Rothe, M. Schunk, P. Sothmann, G. Bretzel, G. Froeschl, C. Wallrauch, T. Zimmer, V. Thiel, C. Janke, W. Guggemos, M. Seilmaier, C. Drosten, P. Vollmar, K. Zwirglmaier, S. Zange, R. Wölfel, M. Hoelscher, Transmission of 2019-NCOV infection from an asymptomatic contact in Germany, New England Journal of Medicine 382 (2020) 970–971.

[3] W.-j. Guan, Z.-y. Ni, Y. Hu, W.-h. Liang, C.-q. Ou, J.-x. He, L. Liu, H. Shan, C.-l. Lei, D. S. Hui, B. Du, L.-j. Li, G. Zeng, K.-Y. Yuen, R.-c. Chen, C.-l. Tang, T. Wang, P.-y. Chen, J. Xiang, S.-y. Li, J.-l. Wang, Z.-j. Liang, Y.-x. Peng, L. Wei, Y. Liu, Y.-h. Hu, P. Peng, J.-m. Wang, J.-y. Liu, Z. Chen, G. Li, Z.-j. Zheng, S.-q. Qiu, J. Luo, C.-j. Ye, S.-y. Zhu, N.-s. Zhong, Clinical Characteristics of Coronavirus Disease 2019 in China, New England Journal of Medicine 382 (2020) 1708–1720.

[4] S. Asadi, N. Bouvier, A. S. Wexler, W. D. Ristenpart, The coronavirus pandemic and aerosols: Does COVID-19 transmit via expiratory particles?, Aerosol Science and Technology 54 (2020) 635–638.

[5] WHO, Coronavirus disease (COVID-2019) situation reports, 2020. URL: `https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports`.

[6] J. Panovska-Griffiths, Can mathematical modelling solve the current Covid-19 crisis?, BMC Public Health (2020).

[7] Z. Wang, Y. Xiao, Y. Li, J. Zhang, F. Lu, M. Hou, X. Liu, Automatically discriminating and localizing COVID-19 from community-acquired pneumonia on chest X-rays, Pattern Recognition (2021).

[8] M. U. Kraemer, C. H. Yang, B. Gutierrez, C. H. Wu, B. Klein, D. M. Pigott, L. du Plessis, N. R. Faria, R. Li, W. P. Hanage, J. S. Brownstein, M. Layan, A. Vespignani, H. Tian, C. Dye, O. G. Pybus, S. V. Scarpino, The effect of human mobility and control measures on the COVID-19 epidemic in China, Science (New York, N.Y.) 368 (2020) 493–497.

[9] S. Chang, E. Pierson, P. W. Koh, J. Gerardin, B. Redbird, D. Grusky, J. Leskovec, Mobility network models of COVID-19 explain inequities and inform reopening, Nature (2020).

[10] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton, M. Monod, Imperial College COVID-19 Response Team, A. C. Ghani, C. A. Donnelly, S. M. Riley, M. A. C. Vollmer, N. M. Ferguson, L. C. Okell, S. Bhatt, Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe., Nature (2020).

[11] A. Zeroual, F. Harrou, A. Dairi, Y. Sun, Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study, Chaos, Solitons and Fractals (2020).

[12] J. Stübinger, L. Schneider, Epidemiology of Coronavirus COVID-19: Forecasting the Future Incidence in Different Countries, Healthcare (2020).

[13] M. Liu, R. Thomadsen, S. Yao, Forecasting the Spread of COVID-19 under Different Reopening Strategies, Scientific Reports 10 (2020).

[14] G. De Nicola, M. Schneble, G. Kauermann, U. Berger, Regional nowand forecasting for data reported with delay: A case study in COVID-19 infections (2020).

[15] N. Oliver, B. Lepri, H. Sterly, R. Lambiotte, S. Delataille, M. De Nadai, E. Letouzé, A. A. Salah, R. Benjamins, C. Cattuto, V. Colizza, N. de Cordes, S. P. Fraiberger, T. Koebe, S. Lehmann, J. Murillo, A. Pentland, P. N. Pham, F. Pivetta, J. Saramäki, S. V. Scarpino, M. Tizzoni, S. Verhulst, P. Vinck, Mobile phone data for informing public health actions across the COVID-19 pandemic life cycle, Science Advances (2020).

[16] C. O. Buckee, S. Balsari, J. Chan, M. Crosas, F. Dominici, U. Gasser, Y. H. Grad, B. Grenfell, M. E. Halloran, M. U. Kraemer, M. Lipsitch, C. J. E. Metcalf, L. A. Meyers, T. A. Perkins, M. Santillana, S. V. Scarpino, C. Viboud, A. Wesolowski, A. Schroeder, Aggregated mobility data could help fight COVID-19, 2020. URL: `https://hhi.harvard.edu/`. doi:`10.1126/science.abb8021`.

[17] Q. Zhang, K. Sun, M. Chinazzi, A. P. Y. Piontti, N. E. Dean, D. P. Rojas, S. Merler, D. Mistry, P. Poletti, L. Rossi, M. Bray, M. E. Halloran, I. M. Longini, A. Vespignani, Spread of Zika virus in the Americas, Proceedings of the National Academy of Sciences of the United States of America (2017).

[18] C. Fritz, G. Kauermann, On the Interplay of Regional Mobility, Social Connectedness, and the Spread of COVID-19 in Germany, arXiv (2020).

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural Message Passing for Quantum Chemistry, in: Proceedings of the 34th International Conference on Machine Learning, PMLR, PMLR, 2017, pp. 1263–1272. URL: `http://proceedings.mlr.press/v70/gilmer17a.html`.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems (2020) 1 – 21.

[21] M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam, P. Vandergheynst, Geometric Deep Learning: Going beyond Euclidean data, IEEE Signal Processing Magazine 34 (2017) 18 – 42.

[22] G. Panagopoulos, G. Nikolentzos, M. Vazirgiannis, Transfer graph neural networks for pandemic forecasting, 2020.

[23] J. Gao, R. Sharma, C. Qian, L. Glass, J. Spaeder, J. Romberg, J. Sun, C. Xiao, Stan: Spatio-temporal attention network for pandemic prediction using real world evidence., arXiv: Social and Information Networks (2020).

[24] S. Deng, S. Wang, H. Rangwala, L. Wang, Y. Ning, Cola-GNN: Cross-location attention based graph neural networks for long-term ILI prediction, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, ACM, 2020. URL: https://doi.org/10.1145/3340531.3411975. doi:10.1145/3340531.3411975.

[25] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. J. Blais, S. O'Banion, Examining covid-19 forecasting using spatio-temporal graph neural networks, ArXiv abs/2007.03113 (2020).

[26] L. Held, M. Paul, Modeling seasonality in space-time infectious disease surveillance data, Biometrical Journal 54 (2012) 824–843.

[27] L. Held, S. Meyer, J. Bracher, Probabilistic forecasting in infectious disease epidemiology: the 13th Armitage lecture, Statistics in Medicine 36 (2017) 3443–3460.

[28] P. Maas, I. Shankar, A. Gros, P. Wonhee, L. Mcgorman, C. Nayak, A. P. Dow, Facebook Disaster Maps: Aggregate Insights for Crisis Response & Recovery, Proceedings ofthe 16th ISCRAM Conference (2019).

[29] G. Bonaccorsi, F. Pierri, M. Cinelli, F. Porcelli, A. Galeazzi, A. Flori, A. L. Schmidth, C. M. Valensise, A. Scala, W. Quattrociocchi, F. Pammolli, Evidence of Economic Segregation from Mobility Lockdown During COVID-19 Epidemic, SSRN Electronic Journal (2020).

[30] L. Lorch, W. Trouleau, S. Tsirtsis, A. Szanto, B. Schölkopf, M. Gomez-Rodriguez, A Spatiotemporal Epidemic Model to Quantify the Effects of Contact Tracing, Testing, and Containment (2020).

[31] D. Holtz, M. Zhao, S. G. Benzell, C. Y. Cao, M. Amin Rahimian, J. Yang, J. Allen, A. Collis, A. Moehring, T. Sowrirajan, D. Ghosh, Y. Zhang, P. S. Dhillon, C. Nicolaides, D. Eckles, S. Aral, Interdependence and the Cost of Uncoordinated Responses to COVID-19, Technical Report, 2020.

[32] A. Galeazzi, M. Cinelli, G. Bonaccorsi, F. Pierri, A. L. Schmidt, A. Scala, F. Pammolli, W. Quattrociocchi, Human Mobility in Response to COVID-19 in France, Italy and UK (2020).

[33] M. Bailey, R. Cao, T. Kuchler, J. Stroebel, A. Wong, Social connectedness: Measurement, determinants, and effects, 2018. doi:10.1257/jep.32.3.259.

[34] Facebook, GeoInsights Help, 2020. URL: https://www.facebook.com/help/geoinsights.

[35] F. Günther, A. Bender, K. Katz, H. Küchenhoff, M. Höhle, Nowcasting the COVID-19 pandemic in Bavaria, Biometrical Journal (2020) bimj.202000112.

[36] R. Koenker, S. Leorato, F. Peracchi, Distributional vs. quantile regression (2013).

[37] S. C. Hora, Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management, Reliability Engineering & System Safety 54 (1996) 217–223.

[38] N. Klein, T. Kneib, S. Lang, A. Sohn, Bayesian structured additive distributional regression with an application to regional income inequality in Germany, Annals of Applied Statistics 9 (2015) 1024–1052.

[39] S. N. Wood, Generalized additive models: an introduction with R, Chapman and Hall/CRC, 2017.

[40] D. Ruppert, M. P. Wand, R. J. Carroll, Semiparametric regression, Cambridge University Press, Cambridge and New York, 2003.

[41] D. A. De Waal, J. V. Du Toit, Automation of generalized additive neural networks for predictive data mining, Applied Artificial Intelligence 25 (2011) 380–425.

[42] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al., Wide & deep learning for recommender systems (2016) 7–10.

[43] N. Umlauf, N. Klein, A. Zeileis, Bamlss: Bayesian additive models for location, scale, and shape (and beyond), Journal of Computational and Graphical Statistics 27 (2018) 612–627.

[44] C. Brás-Geraldes, A. Papoila, P. Xufre, Generalized additive neural network with flexible parametric link function: model estimation using simulated and real clinical data, Neural Computing and Applications 31 (2019) 719–736.

[45] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, G. E. Hinton, Neural Additive Models: Interpretable Machine Learning with Neural Nets, arXiv preprint arXiv:2004.13912 (2020).

[46] D. Rügamer, C. Kolb, N. Klein, A Unified Network Architecture for Semi-Structured Deep Distributional Regression, arXiv preprint arXiv:2002.05777 (2020).

[47] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, arXiv preprint arXiv:1312.6203 (2013).

[48] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Computational Social Networks 6 (2019).

[49] P. Veličković, A. Casanova, P. Lio, G. Cucurull, A. Romero, Y. Bengio, Graph attention networks, 2018. doi:https://doi.org/10.17863/CAM.48429.

[50] W. L. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: NIPS, 2017. URL: https://arxiv.org/abs/1706.02216.

[51] R. V. L. C. Y. L. Annamalai Narayanan Mahinthan Chandramohan, S. Jaiswal, graph2vec: Learning Distributed Representations of Graphs, in: Proceedings of the 13th International Workshop on Mining and Learning with Graphs (MLG), 2017.

[52] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2017. arXiv:1609.02907.

[53] P. Puig, J. Valero, Count data distributions: Some characterizations with applications, Journal of the American Statistical Association (2006).

[54] D. Lambert, Zero-inflated poisson regression, with an application to defects in manufacturing, Technometrics 34 (1992) 1–14.

[55] M. Schneble, G. De Nicola, G. Kauermann, U. Berger, Nowcasting fatal COVID-19 infections on a regional level in Germany, Biometrical Journal (2020) bimj.202000143.

[56] S. N. Wood, Inference and computation with generalized additive models and their extensions, Test (2020).

[57] A. H. Lee, K. Wang, K. K. Yau, Analysis of zero-inflated poisson data incorporating extent of exposure, Biometrical Journal 43 (2001) 963–975.

[58] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M. M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model CNNs, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017. URL: https://doi.org/10.1109/cvpr.2017.576. doi:10.1109/cvpr.2017.576.

[59] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift., in: F. R. Bach, D. M. Blei (Eds.), ICML, volume 37 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2015, pp. 448–456. URL: `http://dblp.uni-trier.de/db/conf/icml/icml2015.html#IoffeS15`.

[60] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[61] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

[62] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in neural information processing systems, 2017, pp. 6402–6413.

[63] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4 (2012) 26–31.

[64] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16 (2016) 785–794. ArXiv: 1603.02754.

[65] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, Advances in neural information processing systems 25 (2012) 2951–2959.