Real-Time Estimation of COVID-19 Infections: Deconvolution and Sensor Fusion

Maria Jahja, Andrew Chin, and Ryan J. Tibshirani

Abstract. We propose, implement, and evaluate a method to estimate the daily number of new symptomatic COVID-19 infections, at the level of individual U.S. counties, by deconvolving daily reported COVID-19 case counts using an estimated symptom-onset-to-case-report delay distribution. Importantly, we focus on estimating infections in real-time (rather than retrospectively), which poses numerous challenges. To address these, we develop new methodology for both the distribution estimation and deconvolution steps, and we employ a sensor fusion layer (which fuses together predictions from models that are trained to track infections based on auxiliary surveillance streams) in order to improve accuracy and stability.

Key words and phrases: COVID-19, nowcasting, deconvolution, sensor fusion.

1. INTRODUCTION

Accurate, real-time estimates of incident infections play a critical role in informing the public health response to the spread of a disease through a population. However, official metrics on disease activity published by traditional public health surveillance systems in the United States do not in fact reflect activity in real-time, as they suffer from some degree of latency due to the way their reporting pipelines are set up and implemented.

With addressing the latency in traditional public health reporting a part of the motivation, the last decade has seen a rise in the development of *digital surveillance* streams in public health. Search and social media trends have constituted much of the focus (e.g., Brownstein, Freifeld and Madoff, 2009; Ginsberg et al., 2009; Salathé et al., 2012; Kass-Hout and Alhinnawi, 2013; Paul and Dredze, 2017). More broadly, *auxiliary surveillance* streams that operate outside of traditional public health surveillance, like online surveys, medical device logs, or electronic medical records, have also received significant attention (e.g., Kass-Hout and Zhang, 2011; Carlson et al., 2013; Viboud et al., 2014; Smolinski et al., 2015; Santillana et al., 2016; Charu et al.,

Maria Jahja is Ph.D. Candidate, Department of Statistics & Data Science, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA (e-mail: maria@stat.cmu.edu). Andrew Chin is Statistical Developer, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA (e-mail: achin23@jhu.edu). Ryan J. Tibshirani is Professor, Department of Statistics & Data Science, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA (e-mail: ryantibs@cmu.edu).

2017; Yang et al., 2019; Ackley et al., 2020; Leuba et al., 2020; Radin et al., 2020).

Auxiliary surveillance can improve not only on the timeliness but also on the accuracy and robustness of traditional public health reporting. Auxiliary data streams have therefore become an integral part of modern systems for disease *nowcasting* (e.g., McIver and Brownstein, 2014; Santillana et al., 2015; Yang, Santillana and Kou, 2015; Farrow, 2016; Jahja et al., 2019; Brooks, 2020), which, put broadly, are used to estimate the contemporaneous value of a signal that will only be fully observed at a later date, using partial or noisy data.

1.1 Surveillance During the Pandemic

During the COVID-19 pandemic, public health surveillance has produced, on one hand, some of the most detailed public health data that the U.S. has ever seen, such as daily, county-level data on reported COVID-19 cases and deaths. It has also, on the other hand, painted an imperfect picture of situational awareness, which created a number of downstream challenges for the public health response. See, e.g., Rosenfeld and Tibshirani (2021) and references therein for an overview of the issues. In this paper, we identify a few issues surrounding COVID-19 case reporting in particular, propose methodology to address them, and implement and evaluate this proposal over eight months of pandemic data.

To give some background, in the early days of the pandemic, a handful of non-govermental groups such as JHU CSSE (Dong, Du and Gardner, 2020) (and also the COVID Tracking Project, the New York Times, and USAFacts) became known as the most trustworthy sources for aggregate public health reporting data on COVID-19 in the U.S. They were founded around the idea of scraping COVID-19 data

published daily on dashboards that are run by local public health authorities (such as state and county departments of public health), which, at the time, provided more accurate and timely data than federal health authorities (probably due to unrecoverable failures at one or more points along the reporting pipeline). In fact, not only in the early days of the pandemic, but throughout, the data published by these groups has been invaluable for decision-makers, modelers, journalists, and the general public; for example, data from JHU CSSE remains the gold standard for COVID-19 case and death forecast evaluation in the COVID-19 Forecast Hub (Reich Lab, 2020), a community-driven repository of forecasts that serves as the official source for forecasting communications by the U.S. CDC.

Turning our focus now to case reporting, JHU scrapes cumulative case numbers that are published daily on local health authority dashboards, and subsequently derives a notion of case incidence based on day-to-day differences in cumulative counts. Note that, by construction, this definition of incidence reflects the number of new COVID-19 cases that are *reported* (to the public) on any given day. Of course, this is not the same as the number of new cases by date tested, specimen collection date, or symptom onset date. Any of the latter options would be more informative (increasingly so) as a definition of incidence; revamping our surveillance systems so that they can directly provide these and other aggregates of interest to the public health response is a critical task for future public health crises.

The reality of the current pandemic: alignment by report date is the only option available, given the data published broadly on local health authority dashboards, hence collected and aggregated by data scrapers. JHU publishes the number of new COVID-19 case reports per U.S. county, daily, at a 1-day lag. However, since report dates can lag behind symptom onset dates by many days (a typical lag is around 5-10, but lags can be up to 30 days or more; see Figure 3), this is actually giving us a glimpse into COVID activity in the recent past, rather than the present.

Importantly, the CDC publishes a de-identified patient-level data set ("line list") on COVID-19 infections (Centers for Disease Control and Prevention, COVID-19 Response, 2020a), which provides a symptom onset date column. In principle, this should allow us to construct a notion of case incidence that is aligned by symptom onset date, but this is not possible in practice, due to two barriers. First, the CDC only publishes updates to the line list monthly (due to the complexity of managing this data set). Second, and more problematically, this line list is fraught with missingness, extending well beyond missingness in the symptom onset column: the *total* number of COVID-19 cases according to this line list (whether the symptom onset date is observed or not) is far less than the total number of cases from JHU (e.g., in early September 2021, the CDC line list reports about 30 million total versus about 40 million from JHU), and some states (such as Texas) appear to missing nearly all of their cases in the line list altogether (see Figure 2).

1.2 Nowcasting by Deconvolution

In what follows in this paper, we use the CDC line list to estimate a delay distribution between symptom onset and report dates, and then use this delay distribution to deconvolve daily numbers of new case reports published by JHU CSSE to estimate daily numbers of new symptomatic infections. Moreover, we train models that track historical trends between past infection estimates and auxiliary signals of COVID-19 activity from Delphi's COVIDcast project (Reinhart et al., 2021), and we fuse together predictions from these models in order to improve the accuracy and robustness of our estimates of new infections for the most recent 10 days (where deconvolution is particularly challenging). An illustration is given in Figure 1.

We focus on estimating new infections in *real-time*, laying out a framework for an operational nowcasting system that is forced to cope with all of the challenges of disease tracking using provisional data. At any given nowcast date t, to estimate the number of symptomatic infections at day $t - k$ (for small values of k, such as $k = 1, 2, \ldots$), we make sure to use data that would have actually been available at t. This not only affects the way we carry out all of our experiments (model training and evaluation), it also leads us to develop some new interesting methodology to deal with the issue of *right truncation* (highlighted in Figure 1 by the blue region). For example, in order to estimate the delay distribution in real-time, we develop a Kaplan-Meier-like procedure to deal with a kind of right censoring that occurs in the line list. We also develop specialized regularization techniques to control the volatility of estimates around the nowcast date in an optimization problem that we solve for real-time deconvolution.

An outline for this paper is as follows. In Section 2, we cover various preliminary details about the problem setup. Retrospective construction of the delay distribution and deconvolution are described in Section 3, whereas real-time estimation is the focus in Section 4. Sensor fusion is covered in Section 5, and extensive evaluations—comparing nowcasts made in real-time to those made retrospectively (using "finalized" data that would have only been available much later), are performed in Section 6. In Section 7, we conclude with a discussion and describe a few directions for future work.

R and Python code for reproducing all figures and results in this paper can be found at https://github.com/cmu-delphi/stat-sci-nowcast.

1.3 Related Work

In the computational epidemiology literature, the term "nowcasting" has been applied to a variety of related but distinct estimation problems. Broadly speaking, what these problems have in common is that they are about real-time estimation of some quantity, based on partial or noisy data. They differ in *what* is being estimated, and whether this
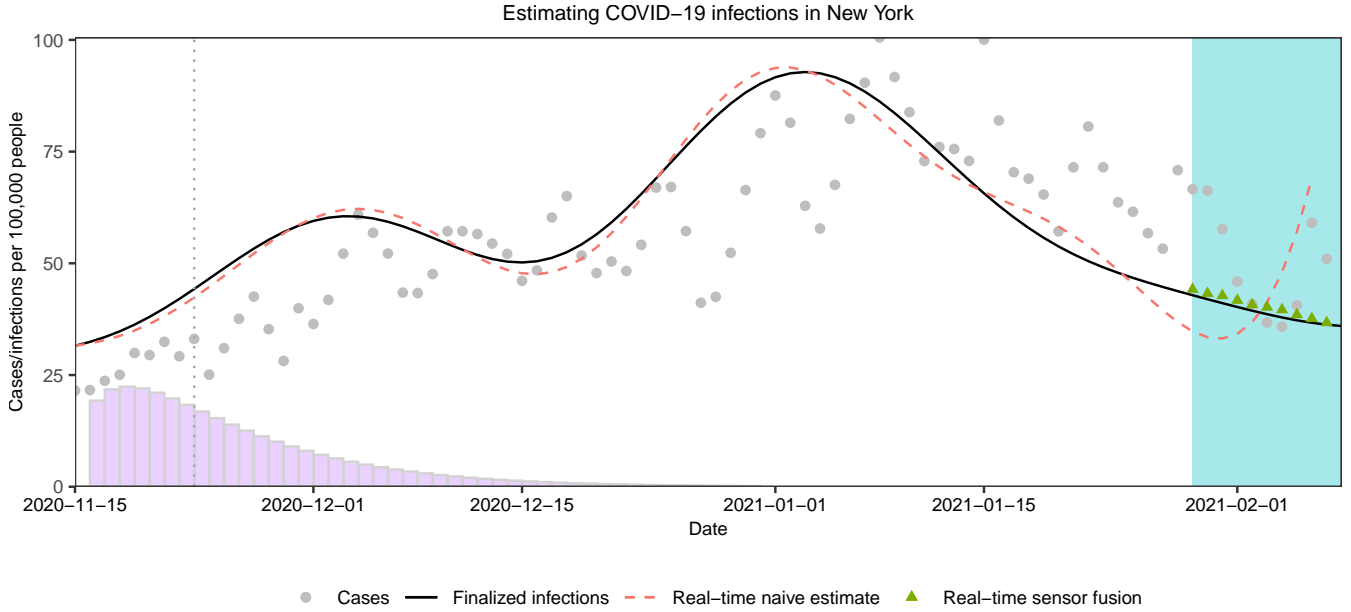
FIG 1. *Illustration of estimating latent infections from reported cases. The dashed red line displays infection rates estimated "naively" in real-time, by directly deconvolving case data up through early February 2021, while the solid black line display infection rates estimated using finalized data from roughly four months afterwards. The blue region on the right-hand side highlights a period in which the real-time estimate deviates substantially from the finalized one, due to the fact that we are lacking sufficient (future) case observations needed to perform a "full" deconvolution. The green triangles represent real-time nowcasts made by sensor fusion, which reduces the volatility of the real-time estimate and tracks the finalized estimate nicely. Lastly, the (scaled) reporting delay distribution estimated at the midpoint of November 2020 is drawn in purple, with the median reporting delay (8 days) marked as a dotted gray line.*

quantity will eventually be fully observed (after enough time has passed) or whether it is latent. Examples in the former non-latent setting, which span applications in influenza, dengue, and COVID-19, include Yang, Santillana and Kou (2015); Farrow (2016); Jahja et al. (2019); Brooks (2020); McGough et al. (2020); Hawryluk et al. (2021).

The latent setting exhibits another degree of diversity within itself. In our work, we target symptomatic COVID-19 infections, which, to be perfectly clear, is a latent time series. Another example along similar lines is Goldstein et al. (2009), who estimate influenza infection incidence via Bayesian deconvolution of mortality data. Meanwhile, other authors might view inferring latent infections as just a stepping stone toward ultimately estimating the instantaneous reproductive number R_t, a key epidemic parameter. Important contributions to the methodology on real-time estimation of R_t include: Bettencourt and Ribeiro (2008), who use a local approximation to the SIR model, and Cori et al. (2013); Thompson et al. (2019), who use a discretization of the renewal equation within a Bayesian framework. For a thorough review and comparison of these methods, see Gostic et al. (2020). The latter paper also discusses in some detail the importance of properly modeling the delay between infection onset and case report, and the issue of right truncation, which, as we will see, are central issues in our paper as well.

The aforementioned methods have been applied and extended to build systems for real-time R_t nowcasting during

the COVID-19 pandemic by Abbott et al. (2020); Systrom, Vladek and Krieger (2020); Chitwood et al. (2021). A key difference between these approaches and ours is that they infer infections through forward-filling: loosely speaking, they convolve forward a candidate estimate of infections, obtain feedback by comparing the result to measured cases, and iterate to refine estimates. This can be effective given accurate prior knowledge, but of course it can be hard to judge the accuracy of prior knowledge in practice. We take a more flexible approach and estimate infections via direct deconvolution. Our approach is nonparametric, but is still fairly simple and computationally efficient. We also focus on fusing in auxiliary sources of information in order to improve real-time accuracy and robustness. We remark that, if estimates of R_t were desired, then these could certainly be inferred as a by-product of our infection nowcasts.

Finally, deconvolution has been extensively studied for many years in many fields, notably signal and image processing, where deconvolution is sometimes called deblurring. As an inverse problem, deconvolution is ill-posed in settings in which the convolution operator is not known exactly or observations are made with noise (Oppenheim and Verghese, 2017). Approaches to overcome this traditionally involve regularization, as in the classical Wiener deconvolution (Wiener, 1964), which stabilizes the inversion using an estimated signal-to-noise ratio. Alternative approaches employ familiar regularization techniques such

as ℓ_1 and ℓ_2 penalities (Taylor, Banks and McCoy, 1979; Debeye and Van Riel, 1990). Most related to our paper is deconvolution using total variation regularization, first proposed by Rudin and Osher (1994), and now a central tool in signal and image processing.

2. PRELIMINARIES

In the remainder of this paper, we develop a framework for estimating the daily symptomatic COVID-19 infection rate (where by "rate" we mean a count per 100,000 people, the standard units in epidemiology), concentrating on infections that will eventually result in a reported COVID-19 case. To be clear on nomenclature: for convenience, we will often abbreviate "symptomatic infection" by "infection" (and so, terms like "infection onset" and "infection rate" should be implicitly interpreted as symptomatic). To estimate infection rates, we deconvolve reported case rates with an estimated symptom-onset-to-case-report delay distribution. To reiterate, we use case data from JHU CSSE (Dong, Du and Gardner, 2020), and to infer the delay distribution, we use a de-identified line list on patient-level infections from the CDC (Centers for Disease Control and Prevention, COVID-19 Response, 2020a).

Auxiliary Indicators. After deconvolution, we improve our infection rate estimates by incorporating a number of contemporaneous signals that track COVID activity—we will also refer to these as *indicators*—which are publicly available through Delphi's COVIDcast API (Reinhart et al., 2021). The five indicators that we consider, described below, provide auxiliary information on COVID-19 outside of traditional public heath reporting. Here and throughout, we abbreviate COVID-like illness by CLI.

1. Change Healthcare COVID (CHNG-COVID): The percentage of outpatient visits that have confirmed COVID-19 diagnostic codes, based on de-identified Change Healthcare medical claims data.
2. Change Healthcare CLI (CHNG-CLI): The percentage of outpatient visits that have COVID-like diagnostic codes, based on the same data.
3. Doctor Visits CLI (DV-CLI): The same definition as CHNG-CLI, but applied to de-identified medical claims data from other health systems partners.
4. COVID Trends and Impact Survey CLI in the community (CTIS-CLIIC): The estimated percentage of people reporting illness in their household or local community, based on Delphi's COVID Trends and Impact Survey (CTIS), in partnership with Facebook.
5. Google searches for anosmia and ageusia (Google-AA): A measure of volume for Google queries related to anosmia or ageusia (loss of smell or taste), from Google's COVID-19 Search Trends data set.

Roughly speaking, we study these particular indicators (ordered roughly from "late" to "early") because conceptually they reflect data measurements that would be made at some period of time in between infection onset and case report to a public health authority, and therefore would be relevant in inferring latent infection rates. More information on these indicators and their underlying data sources is given in Reinhart et al. (2021). For more information on CTIS in particular, see Salomon et al. (2021); and for a study of how these and similar indicators can improve COVID-19 forecasting, see McDonald et al. (2021).

Sensor Fusion. For each of the auxiliary indicators described above, we train a model to estimate latent infection rates from indicator values, using historical data (described in Section 5.1). At each nowcast date, we then use such a model to estimate the latent infection rate from the current indicator value, which gives a total of five estimates (one from each of the five models), along with a sixth estimate coming from an autoregressive model trained on historical estimated infection rates. We will refer these six contemporaneous estimates as *sensors*.

In this paper, we consider (as described in Section 5.3) various methods for combining these estimates into a single estimate of the infection rate, which we will call *sensor fusion* methods. Broadly speaking, sensor fusion is a form of ensembling, which is ubiquitous in in predictive modeling in statistics and machine learning, as it can often help improve both accuracy and robustness. In our particular application, the sensors themselves are constructed from data streams operating outside of traditional public health reporting, which itself contributes an additional important angle in terms of robustness.

2.1 Problem Setup

Estimation Period. For every day t in between October 1, 2020 and June 1, 2021 inclusive (243 days in total), we estimate the symptomatic infection rate at day $t - k$, using only data that would have been as of time t, which in this context we call the *nowcast date*. Estimation of the latent infection rate at time $t - k$ (for positive k) is technically a backcast, though we will not be careful to distinguish this notationally from nowcasting, and will generally refer to this as nowcasting at lag k. We produce estimates for each $k = 1, \ldots, 10$, a total of 10 targets per nowcast date t.

When we say above that nowcasts are made using data that would have been available *as of* a given nowcast date t, we mean that we adhere not to only the real-time availability (latency) of signals at t, but also the *version* of the data published at t—simply put, imagine that we "rewind" the clock to time t and query the API to receive the data that would have been returned then. This is possible becausse the COVIDcast API records and provides access to all historical versions of data, as described in Reinhart et al. (2021). As epidemic data is often subject to revision,

if we train and evaluate models on "finalized" data (that would have been available only at a much later time point) then this can lead to inaccurate conclusions about real-time model performance; see, e.g., McDonald et al. (2021).

Further, it is worth noting that reported case data from JHU is available at a 1-day lag, and we assume that there is at least another 1-day lag between symptom onset and case report (explained in Section 3.2). Hence through real-time deconvolution alone we would be able to make nowcasts at a 2-day lag at the earliest. Making nowcasts at a 1-day lag is possible with sensor fusion, using auxiliary signals with 1-day latency (explained in Section 5). In this sense, sensor fusion is able to improve not only accuracy, but also latency, and buys us 1 extra day.

Geographic Scope. We produce nowcasts at the county resolution, but for computational purposes, we restrict our attention to the 200 U.S. counties with the highest population. We additionally produce estimates for each of the 50 U.S. states. (Some of the methodology that we use for sensor fusion requires a geographical hierarchy, thus using the remaining ≈ 3000 U.S. counties we aggregate these within each state to create "rest-of-state" jurisdictions, and make estimates for these as well, for the purposes or maintaining such a hierachy.)

Evaluation. We evaluate all nowcasts made in between October 1, 2020 and June 1, 2021 inclusive (243 days in total) and at each of the 250 locations in consideration (50 states and the 200 largest counties) against latent infection rate estimates obtained by deconvolving the case rate data available as of August 30, 2021. We will refer to the latter as *finalized* infection rate estimates (as opposed to real-time ones); details are given in Section 3.3.

2.2 Confounding

Estimates of COVID infections obtained by deconvolving reported cases will generally underestimate the true number of infections, because many infections are undetected or untested, and as such, do not appear later on in case reports. If we wanted to estimate the true number of symptomatic infections from case reports, then we would need to have some sense of the fraction of symptomatic infections that go untested. Of course, this only gets more complicated if we extend our consideration to both symptomatic and asymptomatic infections.

Other authors, e.g., Chitwood et al. (2021), have taken the ambitious step of proposing and implementing frameworks with parameters that account for such confounding. However, adjustments for case ascertainment and asymptomatic infections generally rely, at least to some nontrivial extent, on model assumptions (typically, mechanistic ones) that are difficult to substantiate.

We take a different perspective and pose the problem as one of real-time deconvolution only. We seek to answer the question:

Can we estimate—in real-time—the number of new symptomatic COVID-19 infections that will eventually appear in case reports?

Hence, by construction, confounding is not a problem that we even attempt to reconcile (because the target we track, infections that eventually show up in case reports, simply inherits any confounding that would be present in the case reporting stream in the first place).

Our approach can be seen as one that runs in parallel (rather than in contradiction) to an approach that explicitly models and removes the effects of confounding in case reporting. We focus on addressing the deconvolution problem as carefully as possible, with a concern for real-time estimation, and an eye toward using auxiliary signals to improve accuracy and robustness. Estimates of parameters that account for confounding (that comes from other work focused on these aspects) could certainly be applied to our deconvolution estimates post hoc in order to adjust them appropriately; we revisit this idea in the discussion.

Lastly, under an assumption that the confounding acts as a multiplicative bias that changes slowly over time, our real-time infection rate estimates—themselves subject to confounding, as explained above—can be post-processed to derive real-time *approximately unconfounded* estimates of R_t. This is also described in the discussion.

3. RETROSPECTIVE DECONVOLUTION

In this section, we study and fit a convolutional model between infections and reported cases. We adopt a *retrospective* angle here and do not concern ourselves with data availability or versioning issues; this is covered in the next section.

3.1 Convolutional Model

For simplicity, we introduce the convolutional model in just a single location. We denote by y_t the number of new cases that are reported at time t, and by x_t the number of new infections that have onset at time t. Our jumping-off point is the following model:

$$(1) \qquad \mathbb{E}[y_t \,|\, x_s, s \le t] = \sum_{s=1}^{t} \pi_t(s)\, x_s,$$

where for each $s \le t$,

$$(2) \quad \pi_t(s) = \mathbb{P}\big(\text{case report at } t \,|\, \text{infection onset at } s\big).$$

We refer to the probabilities above as *delay probabilities* at time t, and the entire sequence $(\pi_t(s) : s \le t)$, as the *delay distribution* at time t.

The justification for (1), (2) is elementary: to count y_t, we enumerate all infections that ever occurred in the past:

$$y_t = \sum_{s=1}^{t} \sum_{i=1}^{x_s} 1\{\text{the } i^{\text{th}} \text{ infection at } s \text{ gets reported at } t\}.$$

Taking a conditional expectation on both sides above, and using linearity, delivers (1), (2).

In the next subsections, we will describe how to estimate the probabilities $\pi_t(s)$ in (2), and how to use this alongside the observed case reports y_t in order to estimate the latent infections in (1).

3.2 Estimating the Delay Distribution

At the outset, we place the following assumptions on the delay distribution in order to make its estimation (using the CDC line list data, to be described shortly) more tractable.

ASSUMPTION 1. Infections are always reported within $d = 45$ days; that is, $\pi_t(s) = 0$ whenever $s < t - d$.

ASSUMPTION 2. The probability of zero delay is zero; that is, $\pi_t(t) = 0$.

ASSUMPTION 3. The delay distribution is geographically invariant (it is the same for any location).

Assumption 1 is innocuous. The vast majority of pairs of recorded infection dates and report dates in the CDC line list data fall within $d = 45$ days of one another. Assumption 2 is perhaps less innocuous but still fairly minor, and it is a consequence of the fact that a delay of zero (infection date equal to report date) has been used inconsistently in the CDC line list: this could mean a true delay of zero, or it could be a code for missingness.

Assumption 3 is the most noteworthy and troublesome. We do *not* believe it to be true that different locations actually have identical patterns of delay between infections and case reports; conversely, we expect there to be a considerable amount of variability between locations in this regard. While we do allow the delay distribution to change over time (see Figure 3 for evidence for the importance of this), we consider Assumption 3 to be a weakness of our work. However, the *data is simply not there* in the CDC line list to warrant location-specific estimation of the delay distribution (see Figure 2), thus we resort to estimating a nation-wide delay distribution.

Meanwhile, it is worth pointing out that better (location-specific) estimates of the delay distribution could be simply plugged into our deconvolution methodology (detailed in Section 3.3) to yield better estimates of latent infections. This would carry over to all of the real-time methodology for deconvolution and sensor fusion (in Section 4) as well. In other words, a strength of our methodology is that it can treat the delay distribution as an input, and a user (say, a local health official) can replace the default nation-wide delay distribution with a more-informed local one in order to get more-informed local estimates.

In light of Assumptions 1 and 2, we change our notation henceforth, and rewrite (1), (2) as:

$$(3) \qquad \mathbb{E}[y_t \,|\, x_s, s \le t] = \sum_{k=1}^{d} p_t(k)\, x_{t-k},$$

where for $k = 1, \dots, d$,

$$(4) \qquad p_t(k) = \mathbb{P}\big(\text{case report at } t \,|\, \text{onset at } t - k\big).$$

CDC Line List. The CDC provides de-identified patient-level surveillance data on COVID-19 in both public and restricted forms (Centers for Disease Control and Prevention, COVID-19 Response, 2020a,b). The restricted one is made available under a data use agreement. The public line list contains the same patient-level records as the restricted one, but it has geographic details withheld. (There is another publicly available that contains geographic details, but withholds temporal details). We use the public data set[1] in this paper for estimating the delay distribution, since missingness compels us to make nation-wide (rather than location-specific) estimates.

It is worth noting that the line list is itself provisional and subject to revision. Furthermore, the CDC only publishes updates to the line list monthly. In this paper, for simplicity, we use a single version of the CDC line list—released on September 9, 2021—to construct all delay distributions. Nonetheless, in our real-time nowcasting experiments, we restrict our access to data in this line list that would have been available at each nowcast date t (rows whose report date to the CDC is at most t) to construct delay distribution estimates at t. This is highly nontrivial, due to bias induced by truncation of data after t (see Section 4.2).

Missing Values. The CDC line list (both public and restricted data sets) is subject to a high degree of missingness. Such missingness manifests itself in a variety of ways. For the public line list published on September 9, 2021:

- it has 29,851,450 rows, compared to 39,365,080 cumulative cases reported by JHU CSSE on September 9, 2021;
- 8.64% of rows are missing the case report date (the `cdc_report_dt` column);
- 53.6% of rows are missing the symptom onset date (the `onset_dt` column);
- of all rows in which symptom onset date is present, the case report date is also present, but when a report date is missing in practice it sometimes gets filled in with the onset date, clouding the interpretation of a zero delay.[2]

Due to the last point, we exclude zero in the construction of all delay distribution estimates, in what follows.

---

[1] The CDC does not take responsibility for the scientific validity or accuracy of methodology, results, statistical analyses, or conclusions presented.

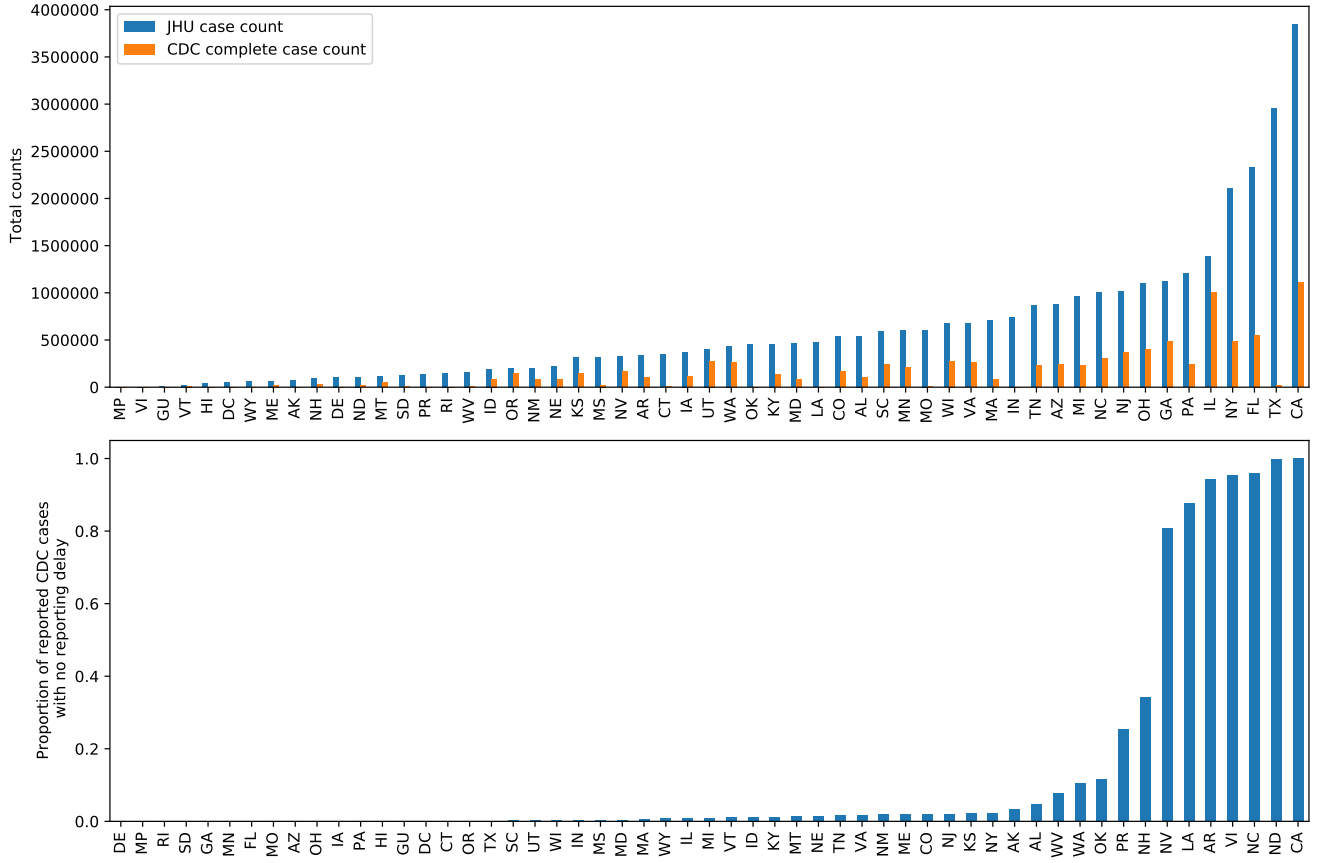[2] Confirmed by personal communication with the CDC.

FIG 2. *Top: cumulative case count per state on June 1, 2021, as reported by JHU CSSE, compared to the complete case count (where both onset date and report date are observed) per state up through the same date, in the CDC restricted line list. Most states have less than 50% of the cases appear in complete form in the line list, and some (e.g., Texas) have almost none at all. Bottom: proportion of complete cases with zero delay per state in the same line list data. There is very wide variation between these proportions.*

The restricted line list is no better with respect to such missingness, exhibiting nearly exactly the same patterns as those described above. It does additionally provide geographic details, which allows us to examine how missingness is dispersed across different locations. Figure 2 displays results to this end, using the restricted line list released on October 12, 2021. The top panel shows that there is a high degree of missingness in complete case counts (those with both onset date and report date observed) in most states, often well over 50%, and moreover, missingness is far from uniform at random: e.g., Texas has barely any of its cases present in the line list. The latter observation is why we resort to estimating nation-wide delay distributions, in what follows.

The bottom panel in the figure shows that there is also a high degree of heterogeneity in the fraction of complete cases with zero delay (between onset date and report date) across states. Some states (e.g., California) have zero delays for nearly all of their complete cases, while others (e.g., Delaware) have zero delays for none of their complete cases, suggesting that the practice of setting a missing report date equal to the associated onset date is highly in-

consistent between states. This only further corroborates the decision to exclude zero delays from the data set when estimating the delay distribution.

Delay Distribution Estimation. From the public line list, we estimate the delay distribution at each time t, namely the probabilities in (4) for $k = 1, \ldots, d$, using the empirical distribution of all lags, excluding zero, between complete onset and report dates, for all onset dates falling in $[t - 2d + 1, t]$. Then, we fit a gamma density to the empirical distribution by the method of moments, and discretize the resulting density over the support $\{1, \ldots, d\}$. For concreteness, this procedure is described in Algorithm 1.

We use only "recent" pairs of onset and report dates at time t (whose onset date lies in $[t - 2d + 1, t]$) in order to adapt to the nonstationarity in reporting delays over time. The top panel in Figure 3 plots quantiles of the estimated delay distribution from Algorithm 1, as t ranges from June 1, 2020 to June 1, 2021. We see sharp drops in all quantiles during the first half of this period, and then a more gradual decline over time. The bottom panel in the figure gives a qualitative sense of how the delay distribution estimates change in shape over time.

Algorithm 1: Delay distribution estimation, retrospective

Input: Time t, support size d, window size $w = 2d$, line list $\mathcal{D}$ with onset dates a_i and report dates b_i.

Output: Estimated delay probabilities $\hat{p}_t(1), \ldots, \hat{p}_t(d)$.

1 Find all pairs in $\mathcal{D}$ with onset dates within a recent time window:
  $I_t = \{i : a_i \in (t-w, t]\}$.

2 Compute the empirical distribution of lags $1, \ldots, d$ among these pairs:

$$\bar{p}_t(k) = \frac{|\{i \in I_t : b_i - a_i = k\}|}{\sum_{\ell=1}^{d} |\{i \in I_t : b_i - a_i = \ell\}|}, \quad k = 1, \ldots, d.$$

3 Fit a gamma density to $\bar{p}_t(1), \ldots, \bar{p}_t(d)$ using the method of moments (matching the mean and variance).

4 Discretize this gamma density to the support set $\{1, \ldots, d\}$, call the result $\hat{p}_t(1), \ldots, \hat{p}_t(d)$, and return these probabilities.

3.3 Defining Ground Truth

Given the estimated delay distributions over time from the previous subsection, we now describe how to estimate latent infections in the model (3). In short, we will solve one large optimization problem to perform deconvolution. To define the best possible retrospective estimates of latent infections over the period October 1, 2020 to June 1, 2021, which we will treat as *ground truth* in what follows (in the sense that they will be the point of comparison for all of our real-time estimates), we will perform deconvolution over a wider time period than the previously specified one in order to avoid any bias issues at the boundaries (where there is insufficient data for accurate deconvolution; more details are provided in the next section): our retrospective deconvolution runs from May 1, 2020 to August 28, 2021, a period we denote by $\mathcal{T}$, and uses case data published on August 30, 2021.

For location ℓ, denote by $y_{\ell,t}$ and $x_{\ell,t}$ the number of new cases reported and number of new infections that onset at time t, respectively, per 100,000 people. Note that $y_{\ell,t}, x_{\ell,t}$ obey (3), (4), because we have just rescaled the underlying counts here by a constant (in order to put them on the scale of rates), and recall, we assume that all locations have the same delay distribution (Assumption 3).

Given the delay distribution estimates from Algorithm 1, $\hat{p}_t = (\hat{p}_t(1), \ldots, \hat{p}_t(d))$ for $t \in \mathcal{T}$, we estimate the full vector $x_\ell = (x_{\ell,t})_{t \in \mathcal{T}}$ of latent infection rates across time, separately for each location ℓ, by solving the problem:

$$(5) \quad \underset{x_\ell}{\text{minimize}} \sum_{t \in \mathcal{T}} \left(y_{\ell,t} - \sum_{k=1}^{d} \hat{p}_t(k) \, x_{\ell,t-k} \right)^2 + \lambda \| D^{(4)} x_\ell \|_1,$$

where $D^{(4)}$ is a matrix such that $D^{(4)} v$ gives all 4th-order differences of a vector v, and $\| \cdot \|_1$ is the ℓ_1 norm. Problem (5) could be called a trend-filtering-regularized least
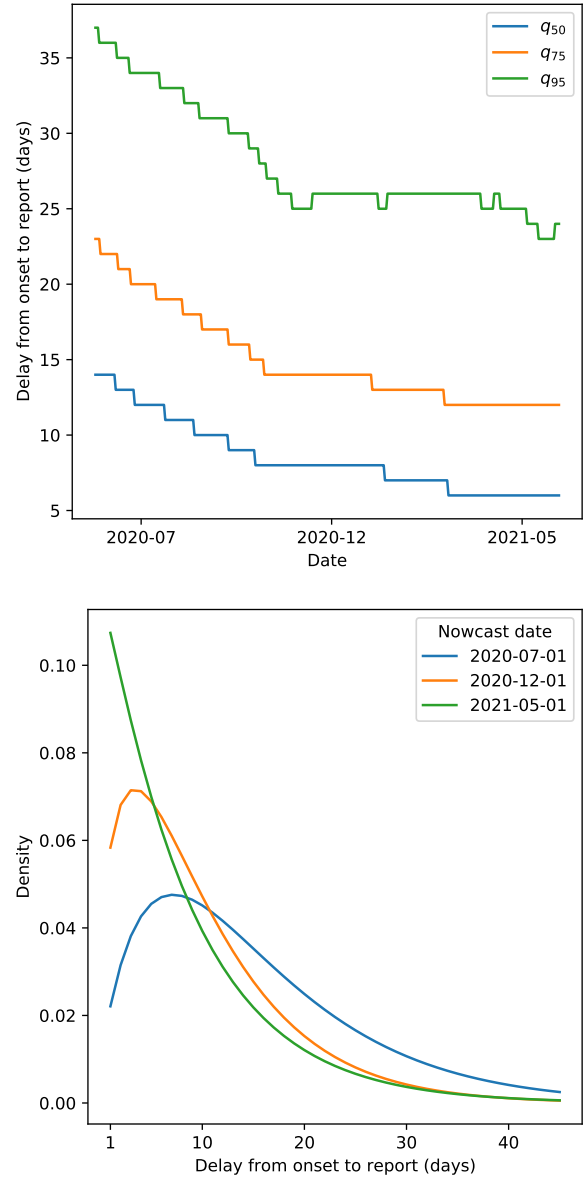

FIG 3. *Top: quantiles of the estimated delay distribution returned by Algorithm 1 at the levels 50%, 75%, and 95%, as t varies from June 1, 2020 to June 1, 2021. Bottom: estimated delay distributions overlaid for three nowcast dates within the same time interval.*

squares deconvolution problem. We solve it (as well as all related optimization problems in this paper) numerically with an adaption of the ADMM algorithm of Ramdas and Tibshirani (2016), detailed in Appendix A.

The solution $\hat{x}_\ell$ in problem (5) takes the form of a cubic piecewise polynomial (discrete spline) with adaptively chosen knots (Tibshirani, 2014, 2020). The tuning parameter $\lambda \geq 0$ controls its complexity, and we choose it using 3-fold cross-validation: we hold out every third value from training, and impute it by the average of the neighboring trained estimates; to compute the validation error, we reconvolve the full vector of imputed infections and measure against observed cases.
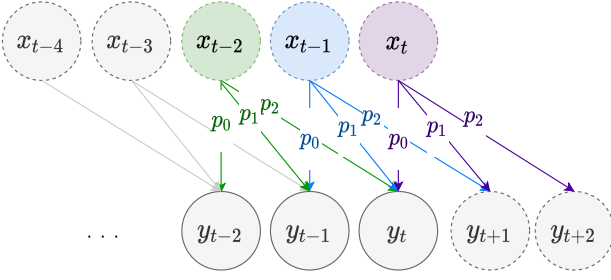
FIG 4. *Illustration of right truncation with a delay distribution of length 3 (which is taken to be stationary for simplicity). At the nowcast time t, some "part" of the latent signal x_t will appear in y_{t+1}, y_{t+2}; likewise, some "part" of x_{t-1} will appear in y_{t+1}.*

4. REAL-TIME DECONVOLUTION

Real-time deconvolution refers to the the task of deconvolving case reports observed up until time t to estimate latent infections up until t, repeatedly, as t marches over the period of interest. We are particularly focused on estimating recent latent infections—nowcasting at a k-day lag, which means estimating at t the latent infection rate at time $t - k$.

Compared to retrospective deconvolution, real-time deconvolution differs in two important ways. The first is that we are forced to work with provisional case data, subject to revision at times in the future, as discussed earlier in Section 2.1. All of our experiments in what follows use properly-versioned data that would have been available as of the nowcast date. We use the notation $y_{\ell,s}^{(t)}$ to reflect the reported case rate in location ℓ at time s as of time t. Reported case data from JHU is available at a 1-day lag and therefore, as of time t, we only observe $y_{\ell,s}^{(t)}$ up through $s = t - 1$ (we use analogous superscript notation for all auxiliary signals and estimates). This means we can only produce deconvolution estimates $\hat{x}_{\ell,s}^{(t)}$ up through $s = t - 2$ (recall we exclude zero delays, in Assumption 2).

The second issue of note, in real-time deconvolution, is *right truncation*: in nowcasting at lag k, where k is small (compared to d), we are only able to carry out a "partial" deconvolution, as much of the needed information would come from case reports occurring in the future, past time the nowcast date t. Figure 4 gives an illustration. Thus, if we simply performed real-time deconvolution by solving the problem analogous to (5), using data that would have been available at time t,

$$(6) \quad \underset{x_\ell^{(t)}}{\text{minimize}} \sum_{s<t} \left(y_{\ell,s}^{(t)} - \sum_{k=1}^d \hat{p}_s^{(t)}(k) \, x_{\ell,s-k}^{(t)} \right)^2 + \lambda \big\| D^{(4)} x_\ell^{(t)} \big\|_1,$$

then we would find that the solution $\hat{x}_\ell^{(t)} = (\hat{x}_{\ell,s}^{(t)} : s < t)$ has highly volatile components for s close to t.

The problem does not stop there; the truncation of data after the nowcast time t also affects estimation of the delay distribution itself. Most rows in the line list with an onset date of $s = t - k$, for small k, will only have a report date (and thus not appear in the line list) until after time t. This means that the estimate $\hat{p}_s^{(t)}$ of p_s given by the empirical distribution of all available line list data, with report date less than t, will be biased toward smaller lag values (i.e., it will place too little weight on larger lag values).

In the next two subsections, we work through each of these truncation issues in turn, by incorporating extra regularization around the right boundary into the criterion in (6), and estimating the delay distribution from truncated data using a Kaplan-Meier-like approach.

4.1 Incorporating Extra Regularization

We consider two forms of extra regularization to dampen the variability of trend filtering estimates toward the right boundary.

Natural Trend Filtering. A natural cubic spline places additional regularity on top of the cubic spline, by maintaining that the function be linear beyond the left and right boundary points of the underlying domain. Natural trend filtering proceeds in a similar vein, but operating in the space of discrete splines; see Tibshirani (2020). Transporting this idea over to our real-time deconvolution problem (6), and applying it to the right boundary only, gives:

$$(7) \quad \underset{x_\ell^{(t)}}{\text{minimize}} \sum_{s<t} \left(y_{\ell,s}^{(t)} - \sum_{k=1}^d \hat{p}_s^{(t)}(k) \, x_{\ell,s-k}^{(t)} \right)^2 + \lambda \big\| D^{(4)} x_\ell^{(t)} \big\|_1$$

$$\text{subject to } x_t^{(\ell)} - 2x_{t-1}^{(\ell)} + x_{t-2}^{(\ell)} = 0.$$

The left and middle panels of Figure 5 demonstrate the improvement that the additional constraints in (7) can have on the boundary estimates, particularly during periods of dynamic change in the underlying case trajectories.

Tapered Smoothing. The right truncation phenomenon is not a binary one and there is increasingly less and less information available for deconvolution as we move the time index s up toward the nowcast date t. Therefore, we design a second penalty to add to the criterion in (7) to gradually increase the amount of regularization accordingly:

$$(8) \quad \underset{x_\ell^{(t)}}{\text{minimize}} \sum_{s<t} \left(y_{\ell,s}^{(t)} - \sum_{k=1}^d \hat{p}_s^{(t)}(k) \, x_{\ell,s-k}^{(t)} \right)^2 + \lambda \big\| D^{(4)} x_\ell^{(t)} \big\|_1 + \gamma \big\| W^{(t)} D^{(1)} x_\ell^{(t)} \big\|_2^2$$

$$\text{subject to } x_t^{(\ell)} - 2x_{t-1}^{(\ell)} + x_{t-2}^{(\ell)} = 0,$$

where $D^{(1)} v$ gives the first-order differences of a vector v, and $W^{(t)}$ is a diagonal matrix that is supported on the last
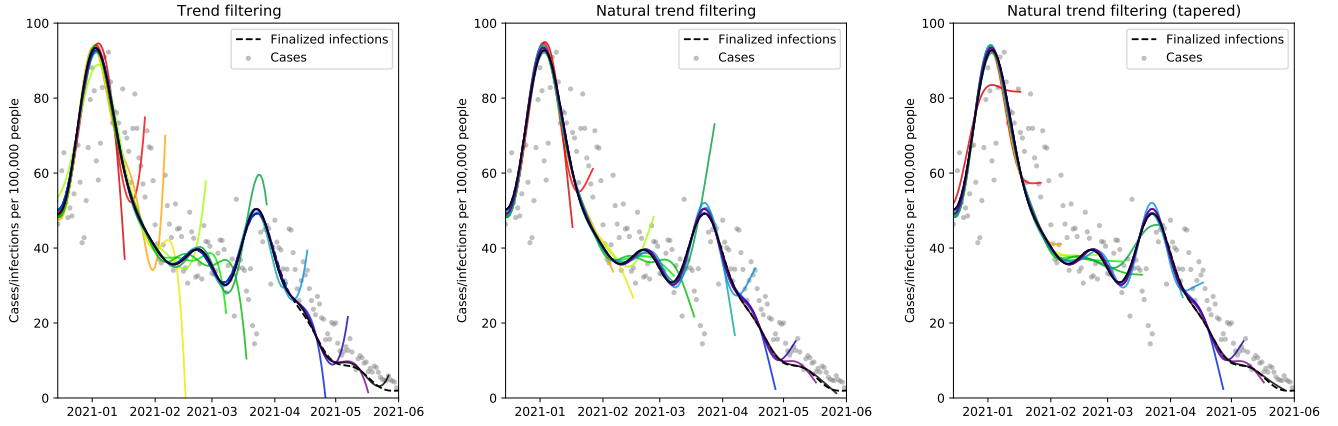
FIG 5. *Comparison of boundary behavior for real-time deconvolution in New York, displayed for a sample of different nowcast dates (where each colored curve traces out the deconvolution estimates for a different nowcast date). The black dashed line indicates finalized infections, estimated roughly three months after June 1, 2021.*

d diagonal entries, these being (in reverse order, starting with the last entry):

$$\frac{1}{\sqrt{\hat{F}_{t-1}^{(t)}(k)}}, \quad k = 1, \ldots, d,$$

where $\hat{F}_{t-1}^{(t)}$ is the cumulative distribution function (CDF) corresponding to the estimated delay distribution $\hat{p}_{t-1}^{(t)}$ at the most recent time $t-1$. The parameter $\gamma \geq 0$ controls the strength of the additional "tapered" penalty in (8), and we tune λ, γ with a two-stage cross-validation procedure:

1. fix $\gamma = 0$, and tune λ using 3-fold cross-validation, as before;
2. fix λ at the value in Step 1, and tune γ using 7-fold forward-validation: for $s = t-2, \ldots, t-8$, we solve the deconvolution problem with a working nowcast date of s, linearly extrapolate to impute an estimate at $s+1$, and then we reconvolve the solution vector along with this imputed point and measure error against observed cases at time $s+1$; the validation error is obtained by averaging these errors over the iterations $s = t-2, \ldots, t-8$.

Figure 6 displays the effect of varying γ on the solution in (8), for a particular deconvolution example, to give a qualitative sense of the role of the tapered penalty. Furthermore, the right panel in Figure 5 demonstrates the benefit this penalty can provide in nowcasting.

Lastly, and importantly, Figure 7 quantifies the improvement offered by the additional regularization mechanisms, in terms of mean absolute error (MAE) measured against finalized infections in nowcasting at a k-day lag, for each $k = 2, \ldots, 10$. This is averaged over all locations and every 10th nowcast date in the evaluation set. We see a considerable improvement in both the natural trend filtering and tapered smoothing modifications, with the biggest improvement occurring when the two are combined as in (8), and hence we stick with this framework in what follows.

4.2 Adjusting the Delay Distribution for Truncation

Now we propose an iterative adjustment to the empirical distribution of truncated line list data in order to overcome the truncation bias. To develop intuition, we first describe the problem using a simple abstraction, formulate a general solution, and then we translate this back over to our particular setting.

KM-Adjustment Under Truncation. Suppose p is a distribution that is supported on $\{1, \ldots, d\}$, and we observe independent random draws that we can partition into two sets: $\mathcal{D}_1$ and $\mathcal{D}_2$, where $\mathcal{D}_2$ contains draws from p and $\mathcal{D}_1$ contains draws from p conditional on the random variable lying in $[1, z_1]$, for a fixed $z_1 \in \{1, \ldots, d\}$. Denote by $\hat{p}_{\mathcal{D}}$ the empirical distribution based on a data set $\mathcal{D}$. Clearly $\hat{p}_{\mathcal{D}_2}$ is unbiased for p, but $\hat{p}_{\mathcal{D}_1}$ is generally biased (it always places zero mass above z_1), and thus the pooled estimate $\hat{p}_{\mathcal{D}_1 \cup \mathcal{D}_2}$ would be biased as well.

To build a more informed estimate based on the pooled sample, the intuition is as follows. First, observe that the only way we can estimate $p(k)$ for $k > z_1$ is by using $\mathcal{D}_2$. Then, this gives an estimate of $S(z_1) = \sum_{k > z_1} p(k)$, the survival function of p at z_1, and we can estimate $p(k)$ for $k \leq z_1$, denoting $Z \sim p$, by observing that

$$p(k) = \mathbb{P}(Z = k \mid Z \leq z_1)(1 - S(z_1)).$$

where we estimate $\mathbb{P}(Z = k \mid Z \leq z_1)$ using the empirical distribution over the set $\mathcal{D}_1 \cup \mathcal{D}_2 \cap [1, z_1]$. In other words, we construct our distribution estimate $\bar{p}$ using two steps:

1. define $\bar{p}(k) = \hat{p}_{\mathcal{D}_2}(k)$ for $k > z_1$, and also $\bar{S}(z_1) = \sum_{k > z_1} \bar{p}(k)$;
2. define $\bar{p}(k) = \hat{p}_{\mathcal{D}_0}(k)(1 - \bar{S}(z_1))$ for $k \leq z_1$, where we let $\mathcal{D}_0 = \mathcal{D}_1 \cup \mathcal{D}_2 \cap [1, z_1]$.

We can readily generalize the above to a setting in which we observe N data sets, with varying levels of truncation:

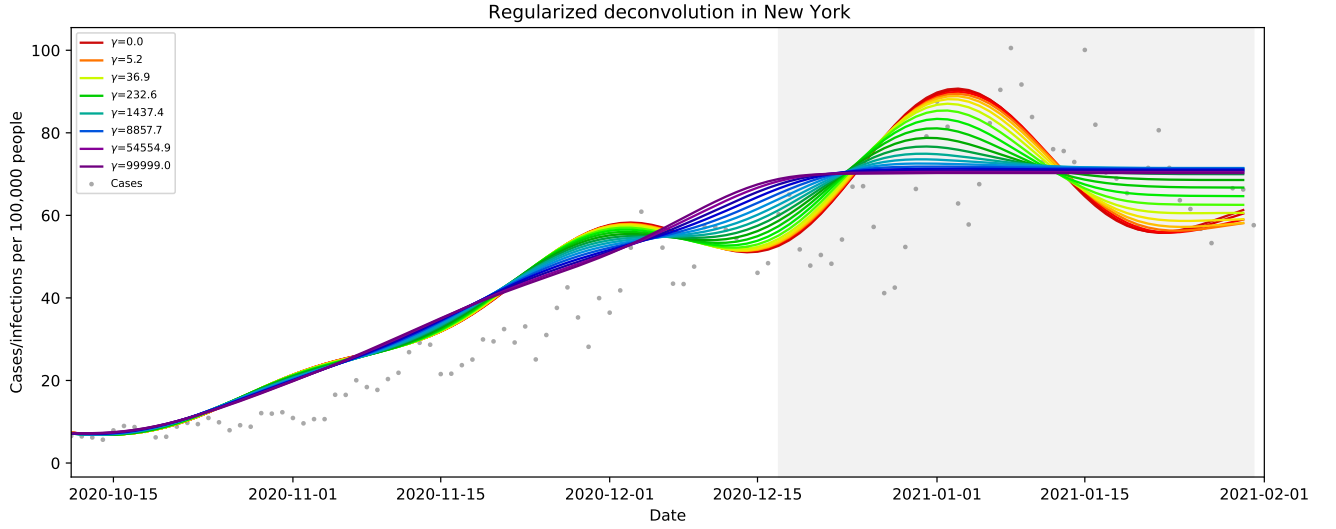$$(9) \qquad \mathcal{D}_i \text{ contains draws } Z \sim p \mid Z \leq z_i, \ i = 1, \ldots, N,$$

FIG 6. *Effect of the tapered smoothing penalty, as we vary the corresponding tuning parameter γ, for a single real-time deconvolution example with on nowcast date February 1, 2021. The gray region highlights the components on which the tapered smoothing penalty acts.*
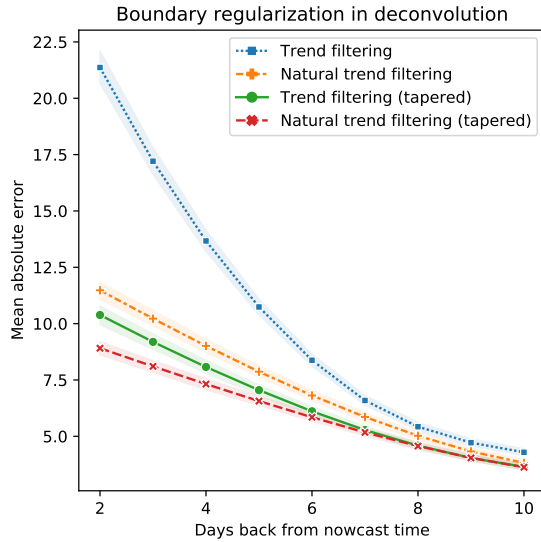

FIG 7. *Comparing regularization approaches by MAE for nowcasting (the shaded bands here and henceforth, in all MAE figures, correspond to 95% bootstrap confidence intervals.) Both approaches for additional regularization give a huge improvement on trend filtering. The biggest improvement comes from combining the two approaches.*

where $1 \le z_1 < \cdots < z_N = d$, and we set $z_0 = 0$ for notational simplicity. To construct an estimate of p based on all the samples, we proceed iteratively as before: first we estimate $p(k)$ for $k > z_{N-1}$ based on the data in $\mathcal{D}_N$, then we estimate $p(k)$ for $k \in (z_{N-2}, z_{N-1}]$ based on data in $\mathcal{D}_{N-1} \cup \mathcal{D}_2 \cap [1, z_{N-1}]$, and so on. Algorithm 2 spells out the procedure in full.

The algorithm just derived may be seen as Kaplan-Meier-like, in the sense that it is motivated by the decomposition

$$p(k) = \mathbb{P}(Z = k \mid Z \le z_i)(1 - S(z_i)), \ \ k \in (z_{i-1}, z_i].$$

Algorithm 2: Distribution estimation under sequential truncation

Input: Data sets and truncation limits $\mathcal{D}_i$ and z_i, for $1, \ldots, N$, as in (9).

Output: Estimated probabilities $\bar{p}(1), \ldots, \bar{p}(d)$.

1 Initialize $\bar{S}(d) = 0$.
2 **for** $i = N, \ldots, 1$ **do**
3   Set $\mathcal{D}_0 = \bigcup_{j=i}^{N} D_j \cap [1, z_i]$.
4   Compute $\bar{p}(k)$, for $k \in (z_{i-1}, z_i]$ based on the empirical distribution of data in $\mathcal{D}_0$ and an estimate of the survival function at z_i:

$$\bar{p}(k) = \hat{p}_{\mathcal{D}_0}(k)(1 - \bar{S}(z_i)), \ \ k \in (z_{i-1}, z_i].$$

5   Compute an estimate of the survival function at z_{i-1}:

$$\bar{S}(z_{i-1}) = \bar{S}(z_i) + \sum_{k \in (z_{i-1}, z_i]} \bar{p}(k).$$

6 **end**
7 Return $\bar{p}(1), \ldots, \bar{p}(d)$.

We use an unbiased plug-in estimate for each term in the product above based on the appropriate data. The Kaplan-Meier estimator has a similar plug-in foundation (Kaplan and Meier, 1958), so we refer to our approach as the *KM-adjusted estimator* of the distribution under truncation.

Application to CDC Line List. Porting the last idea over to the CDC line list, we can use it to estimate the delay distribution at time s using the line list as of time t. Note that if $s < t - d$ then we can still use Algorithm 1, as there is no truncation issue whatsoever. However, if $s \ge t - d$, then we would need to apply the KM-adjusted estimator, because we would be using the rows in the line list whose onset date is at or shortly before s, but are only able to see

Algorithm 3: Delay distribution estimation in real-time

Input: Nowcast time t, working onset time s, support size d, window size $w = 2d$, truncated line list $\mathcal{D}^{(t)}$ with onset dates a_i and report dates b_i such that $b_i < t$.

Output: Estimated delay probabilities $\hat{p}_s^{(t)}(1), \ldots, \hat{p}_s^{(t)}(d)$.

1 **if** $s < t - d$ **then**
2   Return probability estimates from Algorithm 1 (setting $t = s$ and $\mathcal{D} = \mathcal{D}^{(t)}$ in the notation of that algorithm).
3 **end**
4 Set $N = d - (t - s) + 2$.
5 **for** $i = 1, \ldots, N - 1$ **do**
6   Define

$$\mathcal{D}_i = \{b_i - a_i : a_i = s - i + 1\}$$
$$z_i = t - s + i - 2.$$

7 **end**
8 Define $\mathcal{D}_N = \{b_i - a_i : a_i \in (s - w, t - d)\}$ and $z_N = d$.
9 Use Algorithm 2 (applied to $\mathcal{D}_i, z_i, i = 1, \ldots, N$) to compute probability estimates $\bar{p}_t(1), \ldots, \bar{p}_t(d)$.
10 Fit a gamma density to $\bar{p}_t(1), \ldots, \bar{p}_t(d)$ using the method of moments (matching the mean and variance).
11 Discretize this gamma density to the support set $\{1, \ldots, d\}$, call the result $\hat{p}_t(1), \ldots, \hat{p}_t(d)$, and return these probabilities.

those whose report date is at most $t - 1$ (thus would have been available at time t). After making this adjustment to the empirical distribution, we apply gamma smoothing as before. This is detailed in Algorithm 3.

Figure 8 compares the KM-adjusted and naive estimates of the delay distribution, Algorithm 3 versus Algorithm 1 applied directly to $\mathcal{D}^{(t)}$, the line list available at each now-cast date t. In terms of ℓ_1 distance, measured to the finalized delay distribution estimate computed retrospectively (based on the full untruncated line list), and averaged over all nowcast dates in the evaluation period, we see that the KM-adjustment greatly improves the accuracy at all lags $k = 2, \ldots, 10$ (where $k = t - s$, the difference between the nowcast and working onset dates).

4.3 Shortening the Deconvolution Window

Lastly, we investigate shortening the window used in the regularized deconvolution problem (8) so that we use only a window length of w days before t:
(10)

$$\underset{x_\ell^{(t)}}{\text{minimize}} \sum_{s \in [t-w,t)} \left(y_{\ell,s}^{(t)} - \sum_{k=1}^{d} \hat{p}_s^{(t)}(k)\, x_{\ell,s-k}^{(t)} \right)^2 +$$
$$\lambda \big\| D^{(4)} x_\ell^{(t)} \big\|_1 + \gamma \big\| W^{(t)} D^{(1)} x_\ell^{(t)} \big\|_2^2$$

subject to $x_t^{(\ell)} - 2x_{t-1}^{(\ell)} + x_{t-2}^{(\ell)} = 0$,

As we are mainly interested in the components of the solution $\hat{x}_s^{(t)}$ for s close to t, shortening the training window is computationally advantageous and should not change the behavior of the solution very much for s close to t.
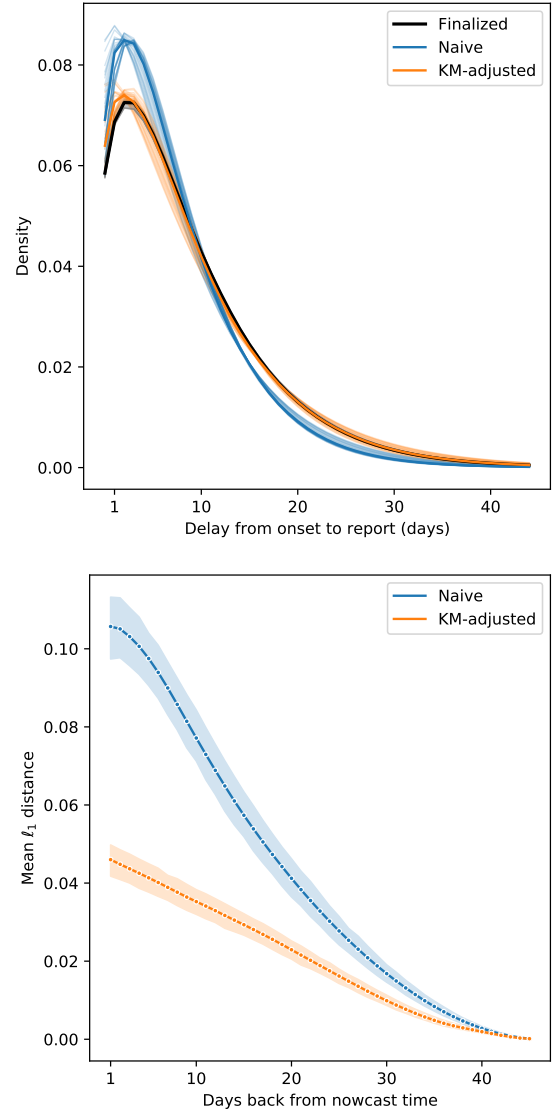

FIG 8. *Top: estimated delay distributions overlaid for all nowcast dates in the month of November 2020, when $s = t - 1$ (working onset date one day before the nowcast date). Bottom: mean ℓ_1 distance to finalized estimate of the delay distribution, as a function of the lag $k = t - s$.*

Figure 9 compares (10) with $w = 2d$, $w = 4d$, and "all-past", which is the original problem (8), in terms of mean absolute error (MAE) measured against finalized infections in nowcasting at a k-day lag, for each $k = 2, \ldots, 10$. This is averaged over all locations and every 10th nowcasting date in the evaluation set. The performance is basically identical for window lengths $2d$ and $4d$, and though all-past may appear to have the slightest advantage, this does not warrant the extra computation, hence in what follows we stick to (10) with a window length $w = 2d$ as our real-time deconvolution estimator.

5. LEVERAGING AUXILIARY SIGNALS

The indicators enumerated in Section 2 have displayed impressive correlations to reported COVID-19 cases (Rein-
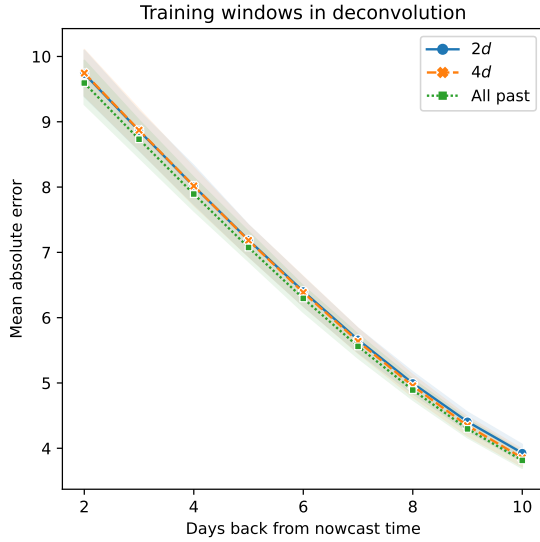
FIG 9. *Comparing window lengths used in regularized deconvolution by MAE for nowcasting. The performance is very similar throughout.*

hart et al., 2021), and moreover, demonstrated an ability to improve the accuracy of case forecasting and hotspot prediction models (McDonald et al., 2021). In this section, we describe how to use each indicator to build a real-time *sensor* that estimates the latent infection rate, and how to fuse such estimates together into a single nowcast.

5.1 Sensor Models

At each prediction time t, for each location ℓ, and for each of the five indicators (abbreviated CHNG-COVID, CHNG-CLI, DV-CLI, CTIS-CLIIC, and Google-AA), we will train a model to predict in real-time latent infections from indicator values. Let $\hat{x}_{\ell,s}^{(t)}$ denote the solution at time s in problem (10), which represents our best estimate of the latent infection rate at time s as of time t from deconvolution of case rates alone.

We use $z_{\ell,s}^{i,(t)}$ to denote the value of indicator i at time s and location ℓ, as of time t. We fit a simple linear model to predict latent infections from indicator values by solving

$$(11) \quad \underset{\beta_0, \beta_1}{\text{minimize}} \sum_{s=t-d}^{t-\tilde{k}_i} w_s^{(t)} \big(\hat{x}_{\ell,s}^{(t)} - \beta_0 - \beta_1 z_{\ell,s}^{i,(t)} \big)^2,$$

which is a weighted linear regression over the time period $[t-d, t-\tilde{k}_i]$, where $\tilde{k}_i = \max\{k_i, 2\}$ and k_i denotes the lag at which indicator i is available. This is:

- $k_i = 1$ for CTIS-CLIIC and Google-AA[3]; and

---

[3]Our treatment of Google-AA is different from the rest. Google's team did not start publishing this signal until September 2020, and the historical latency of this signal was sporadic, but was often longer than 1 week. However, unlike (say) the claims-based signals, revisions are never made after initial publication, and the latency of the signal is not an unavoidable property of the data type, and therefore we use finalized signal values, with a 1-day lag, in our analysis.

- $k_i = 4$ for the claims-based indicators, due to heavy revision or "backfill" over the first several days in the underlying claims data after an outpatient visit date (Reinhart et al., 2021).

Notice that, as defined, $\tilde{k}_i$ is the lag at which *both* the deconvolution estimate of infection rate and auxiliary signal i are available, which is the data we need to fit the linear sensor model (response and covariate data, respectively).

The observation weights in (11) are given by

$$w_{t-k}^{(t)} = \hat{S}_{t-1}^{(t)}(k-1), \;\; k = 1, \ldots, d.$$

Here $\hat{S}_{t-1}^{(t)}$ is the survival function of $\hat{p}_{t-1}^{(t)}$, the estimated delay distribution from the most recent time point $t-1$. We define $\hat{S}_{t-1}^{(t)}(1) = 1$, corresponding to the exclusion of 0-day delays. This scheme upweights the more recent estimates (responses in the regression) of latent infections as they contain more timely information for nowcasting (assuming that the right-truncation bias has been effectively mitigated in the deconvolution step).

Given the solution $\hat{\beta}_{\ell,0}^{i,(t)}, \hat{\beta}_{\ell,1}^{i,(t)}$ in (11), we then define a sensor—which is just a prediction from the fitted linear model—based on indicator i, for time s and location ℓ, as of time t, as:

$$(12) \quad \bar{x}_{\ell,s}^{i,(t)} = \hat{\beta}_{\ell,0}^{i,(t)} + \hat{\beta}_{\ell,1}^{i,(t)} z_{\ell,s}^{i,(t)}.$$

This sensor is available up until $s = t - k_i$. For the CTIS-CLIIC and Google-AA sensors, the lag is $k_i = 1$, smaller than the inherent lag of 2 in the deconvolution estimate.

In brief, each sensor model takes a certain indicator and transforms it—using a location-specific and time-varying mapping—to the scale of local infection rates. While this mapping is simple (based on linear regression), it is also highly nontrivial, as it inherently accounts for geographic biases and nonstationarity.

Finally, in addition to defining sensors based on (11), (12) for each of the five auxiliary sensors, we also define a sixth sensor based on a 3rd order autoregressive model trained on $\hat{x}_\ell^{(t)} = (\hat{x}_{\ell,s}^{(t)} : s < t)$. It is constructed exactly as in (11), (12) (same weights and same training window). Henceforth we abbreviate it AR(3).

5.2 Sensor Missingness

To be clear (11), (12) are to be implicitly understood as performed over observed (non-missing) indicator values. If an indicator value is missing at a particular location and time, then we drop it from the training set in (11), and do not produce a corresponding sensor value in (12). For a summary of missingness in the sensors, see Figure 10.

In general, an indicator will be missing when there is insufficient underlying data (from surveys, medical claims, etc.) to form a reliable signal value at a given location and time. However, the situation is different for the Google-AA indicator: here missingness occurs because the COVID-19
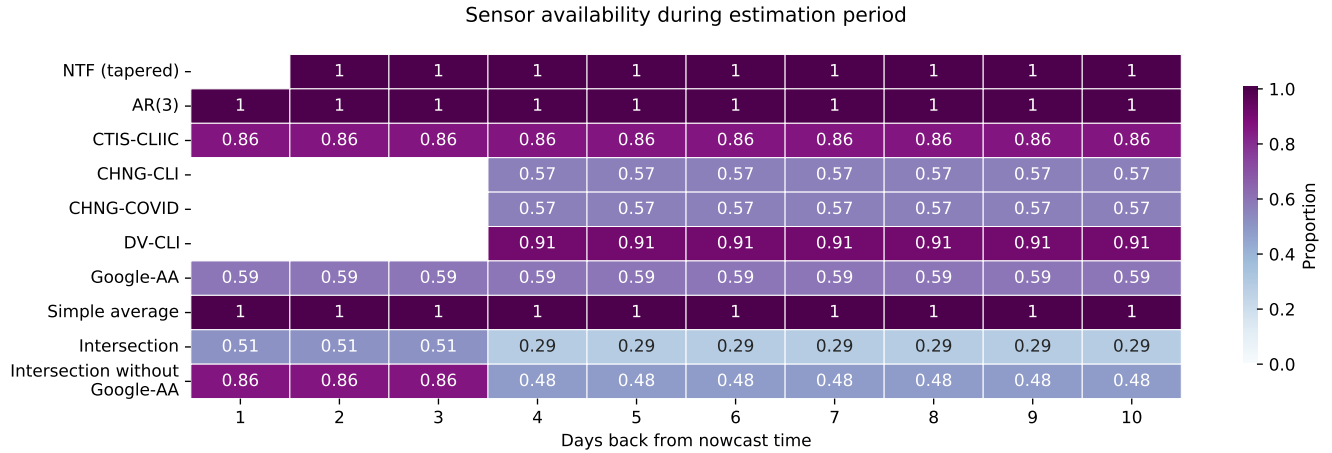
Sensor availability during estimation period

	1	2	3	4	5	6	7	8	9	10
NTF (tapered)		1	1	1	1	1	1	1	1	1
AR(3)	1	1	1	1	1	1	1	1	1	1
CTIS-CLIIC	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
CHNG-CLI				0.57	0.57	0.57	0.57	0.57	0.57	0.57
CHNG-COVID				0.57	0.57	0.57	0.57	0.57	0.57	0.57
DV-CLI				0.91	0.91	0.91	0.91	0.91	0.91	0.91
Google-AA	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
Simple average	1	1	1	1	1	1	1	1	1	1
Intersection	0.51	0.51	0.51	0.29	0.29	0.29	0.29	0.29	0.29	0.29
Intersection without Google-AA	0.86	0.86	0.86	0.48	0.48	0.48	0.48	0.48	0.48	0.48

Days back from nowcast time

FIG 10. *Proportion of observed (non-missing) values over the evaluation period from October 1, 2020 to June 1, 2021, and over all locations, as a function of lag $k = 1, \ldots, 10$. (NTF refers to the real-time deconvolution estimator, and simple average refers to the sensor fusion method that averages all available sensors.) The bottom two rows reflect the intersection of location-time pairs for which all data—deconvolution estimates and sensors—are available for that given lag, with and without including the Google-AA sensor, since this sensor has a large amount of individual missingness. Each intersection at each given lag k is restricted to data whose latency is not greater than k. For example, the bottom leftmost cell computes the porportions of locations and dates at which AR(3), CTIS-CLIIC, and the simple average are concurrently available.*

search trends data set is released after using a differential privacy layer (Bavadekar et al., 2020), and a missing value means that the level of noise added for privacy protection is high compared to the search count. Therefore we impute missing Google-AA signal values by zeros in our analysis; we do this unless the Google-AA signal was missing for a particular location and *all* times in the evaluation period, in which case we leave it as missing for this location entirely.

5.3 Sensor Fusion

Sensor fusion (SF), broadly speaking, refers to the process of assimilating data sources, each of which ideally contains complementary information, in order to produce more accurate estimates or predictions. SF falls into the general class of ensemble methods, and the sensors constructed in the previous section can be thought of as base learners, to be subsequently combined.

We consider the following five ensemble methods. In each case, we describe how to form the estimate at time s and location ℓ as of time t. Though not explicitly stated, it is to be implicitly understood that all sensor values are as of time t as well.

1. Simple average: the average of available sensors at time s and location ℓ.
2. Simple regression: the prediction from a linear regression model at time s and location ℓ, fit to available sensors over the training period at location ℓ.
3. Ridge: the prediction from a ridge regression model at time s and location ℓ, fit to available sensors over the training period and over locations j such that j, ℓ lie in the same U.S. state (including the state sensor itself).

4. Lasso: same as in the last item, but using the lasso instead of ridge regression.
5. KF-SF: the Kalman-Filter-inspired method for sensor fusion from Farrow (2016); Jahja et al. (2019), with covariance shrinkage, and operating on the geographical hierarchy within each U.S. state.

Methods 2–5 are trained on the most recent $2d$ time points, and 3–5 are tuned using 7-fold forward validation, where we allow them to choose a lag-specific tuning parameter. Methods 1–2 are "simple" in the sense that for nowcasting at a location ℓ they use sensors from ℓ only. Methods 3–5 are more sophisticated in that they pool information across locations within the same state.

The KF-SF method requires a proper geographical hierarchy and thus we create "rest-of-state" jurisdictions by aggregating the remaining counties (outside of the top 200 counties nationally) within each state, and to run KF-SF, we create an AR(3) sensor at these rest-of-state locations (since one sensor at each location is sufficient). It is worth noting that, as shown in Jahja et al. (2019), KF-SF bears a close connection to ridge in Model 4: it is in fact equivalent to a modified ridge optimization problem that imposes additional linear constraints.

6. EVALUATION

We now evaluate nowcasting performance over all locations and all but every 10th nowcasting date in our evaluation period from October 1, 2020 to June 1, 2021. (We do this because it gives us a "pure" test set, since every 10th nowcasting date was already used to choose the real-time deconvolution methodology in Section 4.) As before, we

compare to finalized estimates of infection rates computed via retrospective deconvolution, as in Section 3.

For the purposes of making fair comparisons, in every analysis (figure) that we present, we only aggregate over the intersection of nowcasts dates and locations at which the particular estimates under consideration—coming from real-time deconvolution, individual sensor models, or sensor fusion—are all available. Abiding by this rule leads us to examine several different ways of stratifying results, as the full intersection is fairly sparse (see the second-to-last row in Figure 10). In particular, we consider the following two dimensions used to define strata:[4]

- inclusion of Google-AA or not;
- inclusion of all claims-based sensors (CHNG-CLI, CHNG-COVID, and DV-CLI) or not.

In what follows, we first examine the performance of individual sensor models and a certain sensor fusion method (the simple average) compared to real-time deconvolution, and then examine the relative performance of the different sensor fusion methods.

6.1 Performance of Sensors and Sensor Fusion

We begin by comparing the MAE of nowcasts from natural trend filtering (NTF) using tapered smoothing, as in (10) (the real-time deconvolution estimator chosen based on the analysis in Section 4) to those from individual sensor models and the simple average sensor fusion method. Despite its simplicity, the simple average appears to be the best-performing sensor fusion method overall (details in the next subsection), and so we stick with it as the de facto sensor fusion method in this subsection. The results here do not include Google-AA; results including Google-AA are shown in Appendix B.

Figure 11 displays the MAE from various methods as a function of lag k. The top and bottom panels do not and do include the claims-based sensors, respectively. In either case, we see that up until lag 6, all sensors outperform the real-time deconvolution estimate from NTF. The simple average of all sensors improves accuracy even further, and achieves the best MAE for all lags up through lag 6. We recall that NTF (with tapered smoothing) itself already provides a huge increase in accuracy over the more naive method for real-time deconvolution given by applying trend filtering without extra boundary regularization (Figure 7). At lag 7, the NTF estimate catches up to about equal accuracy, and then surpasses sensor fusion and all sensors in accuracy at lag 8 and onward. An interpretation for this: right truncation ceases to be a significant problem

---

[4]To be explicit, when we say we do not "include" certain sensors, it means both that we ignore results from their individual sensor models (in computing the common intersection of available nowcast dates and locations), and *also* that we exclude them in running the sensor fusion methods.
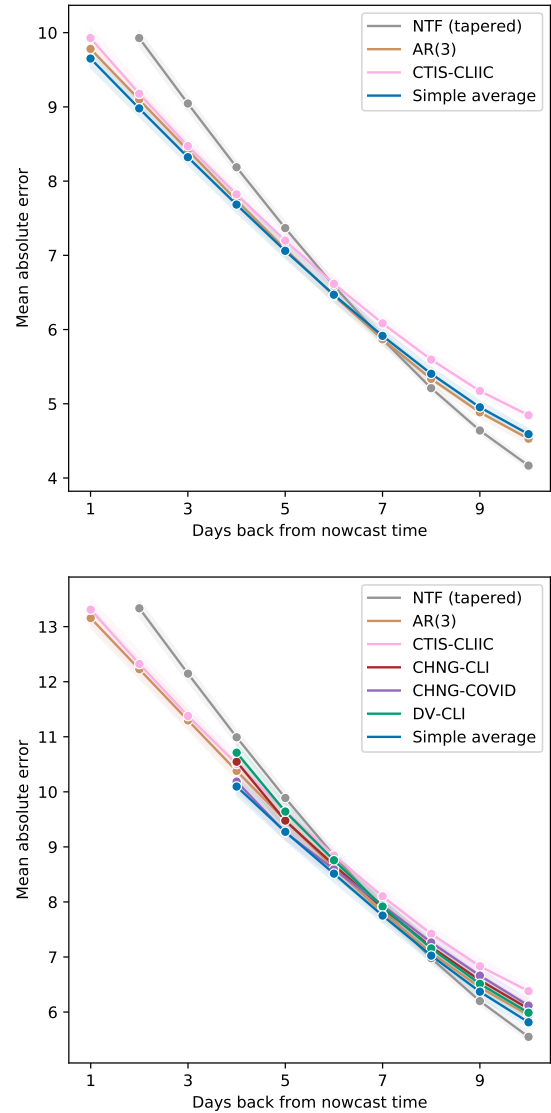

FIG 11. *Comparing NTF to individual sensor models and the simple average sensor fusion method by MAE for nowcasting. The top panel excludes the claims-based sensors, whereas the bottom includes them. For lags smaller than 7, all methods improve upon NTF (with tapered smoothing), with simple average being the best among them.*

past lag 7, and thus we are better off performing deconvolution directly in order to estimate infections more than a week into the past.

Figure 12 displays the empirical distributions of ranks of nowcast errors coming from each method, computed with respect to each other, over common nowcast tasks (defined by a location-date-lag triplet). For example, in a particular nowcast task, we assign a rank of 1 to the method with the smallest absolute error for that nowcast task. The top panel again excludes claims-based signals, and the bottom panel includes them. The striking feature in either panel, particularly the bottom panel, is that the simple average has a highly distinctive distribution of ranks—it is rarely the best method, but never the worst. While this is not
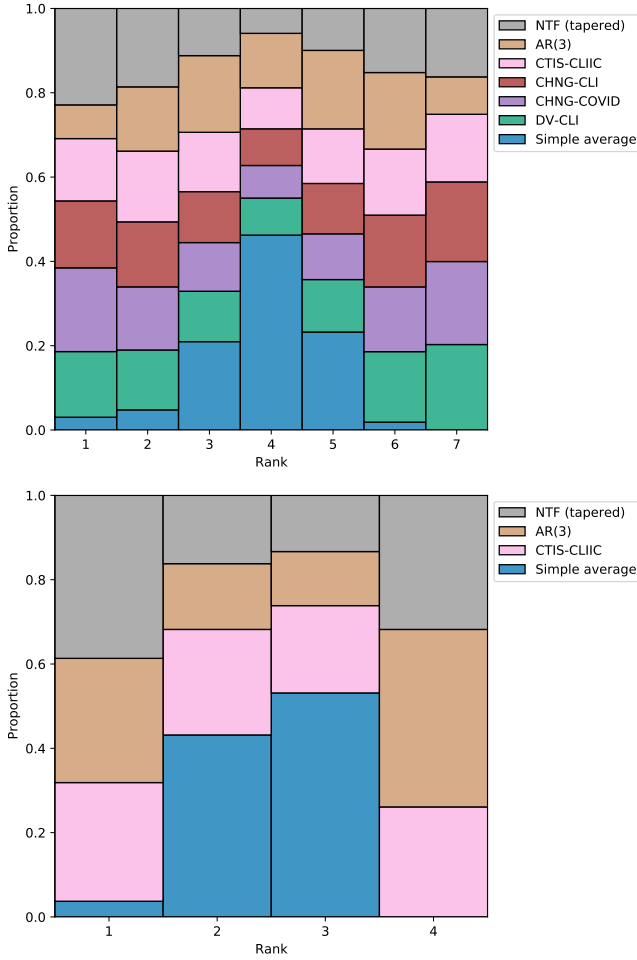
FIG 12. *Comparing NTF to individual sensor models and the simple average sensor fusion method by relative ranks over common nowcast tasks. The top panel excludes all claims-based sensors and considers lags 1–5, whereas the bottom panel includes them and considers lags 4–9 (the first 5 lags at which all methods are available, in either case). The simple average exhibits striking consistency: it is rarely the best, but also never the worst.*

particularly surprising (averaging random variables tends to be variance-reducing, as long as the variables are not too correlated), it also points to a key property of sensor fusion—a certain kind of robustness, beyond accuracy.

6.2 Relative Performance of Sensor Fusion Methods

We now compare the various sensor fusion methods to each other. The results here do not include claims-based signals; results including claims-based signals are shown in Appendix B. Figure 13 displays the MAE of the various sensor fusion estimates, but divided up into three panels, defined by averaging over small, medium, and large states (the figure caption provides more details). Recall that for the lasso, ridge, and KF-SF approaches, a model in a particular county is fit using the sensors from other counties in the same state. Larger states have more pooling of information across locations and present a greater potential for gains in accuracy. We see that the simple average method

is typically the best sensor fusion method at each lag, but for medium and large states, KF-SF catches up with it and is just about as accurate.

Figure 14 displays the relative ranking of sensor fusion methods. The simple average and KF-SF methods appear the most favorable (often the best, and less so the worst), followed by lasso, then ridge, and lastly simple regression (most often the worst).

7. DISCUSSION

In this work, we proposed, implemented, and evaluated a framework for real-time estimation of new symptomatic COVID-19 infections from case reports. At time t, in order to nowcast the infection rate at time $t - k$ (for small values of k, such as $k = 1, 2, \ldots$), the main steps are to:

1. estimate a symptom-onset-to-case-report delay distribution using the most recent data available in a line list provided by the CDC;
2. perform regularized deconvolution on the most recent case data available from JHU CSSE;
3. update models to track recent infection rates from various auxiliary signals (based on COVID-related data from medical insurance claims, online surveys, and Google searches), and fuse together the predictions from these models in order to stabilize recent estimates of infection rates.

In each step, we proposed methodological advances that improved the accuracy of our nowcasts, when measured against finalized infection rate estimates obtained by retrospective deconvolution (using data that would have only been available months later). While using auxiliary signals (step 3) did help in terms of accuracy and robustness, the additional regularization devices that we incorporated into real-time deconvolution (step 2) ended up providing the biggest benefit to accuracy.

To reiterate, we purposely defined our target of estimation to be symptomatic infections that would eventually show up in public health reports, allowing us to focus on developing and testing tools for real-time deconvolution and sensor fusion, with minimal assumptions (e.g., without a mechanistic model for disease spread). Estimating the number of true symptomatic infections at any point in time—whether or not they will appear in case reports—is of course a much harder problem. However, our methodology may be seen as a contribution toward solving this larger problem in real-time; moreover, some simple post hoc corrections could be applied to our real-time estimates in order to adjust for confounding. For example, if $a_{\ell,t}$ is the fraction of untested symptomatic infections in location ℓ at at time t, which (say) is estimated from external data sources, then we could just multiply each element $\hat{p}_{\ell,s}^{(t)}$ of the delay distribution used in (10) by $b_{\ell,t} = 1/(1 - a_{\ell,t})$ in order to estimate *all* symptomatic infections from case
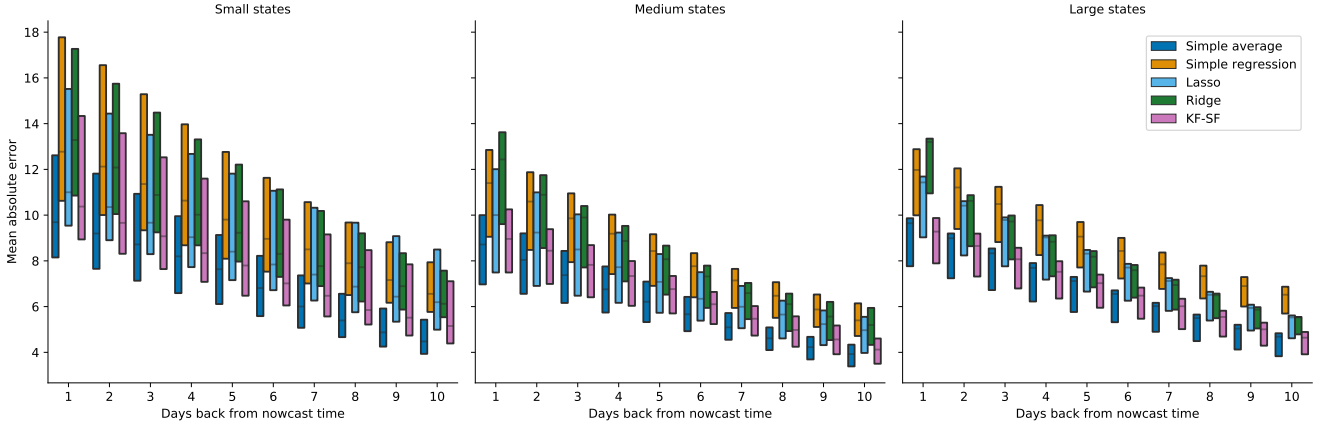
FIG 13. *Comparing sensor fusion methods by boxenplots of nowcasting errors (each box conveys the level 25%, 50%, and 75% quantiles of the absolute error distribution.) The three panels average over small (containing less than 5 locations), medium (between 5 and 14 locations), and large (more than 15 locations) states. Simple average performs generally the best throughout, but KF-SF catches up for medium and large states.*
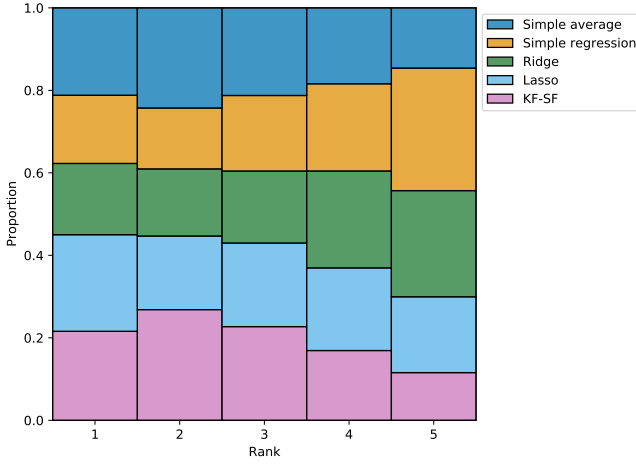

FIG 14. *Comparing sensor fusion methods by relative ranks over common nowcast tasks, and considering only lags 1–5. The simple average and KF-SF methods consistently perform in the top half, while simple regression is most often the worst.*

reports. Due to the way we have set up the deconvolution problem (cross-validating over optimal choices of tuning parameters), this would be essentially equivalent to post-mulitplying the nowcast $\hat{x}_{\ell,s}^{(t)}$ we already produce by $b_{\ell,t}$.

We finish by describing a few directions for future work.

Post Hoc Smoothing. As we saw in Section 6, sensor fusion provides a real-time improvement on pure deconvolution up until about a 10-day lag, and past that point, the deconvolution estimates appear stable enough that sensor fusion becomes unnecessary. While the quantative benefit of sensor fusion for small lags is clear, sensor fusion is also lacking in the following qualitative aspect: its estimates do not always appear visually smooth across time (this is because the sensors themselves need not be smooth over time, and furthermore, sensor fusion may end up using a different subset of sensors at each lag, creating additional

jaggedness). Post smoothing techniques would be worth investigating here, to aid visual consumption.

R_t *Estimation.* The instantaneous reproductive number R_t, the average number of secondary infections at time t generated from a primary infection in the past, is a useful and interpretable parameter that reflects the dynamics of epidemic growth in a population. In the SIR model, the instantaneous reproductive number R_t and growth rate r_t at time t obey the following relationship:

$$R_t \approx 1 + \frac{r_t}{\gamma},$$

where γ denotes the recovery rate in the SIR model. While this is well-known in the literature on mathematical modeling of epidemics (and is exact under local exponential growth; see, e.g., Wallinga and Lipsitch (2007)), its use in the presence of confounding seems to be underexplored and potentially undervalued. If I_t denotes the number of new infections at t, then using a simple discrete difference approximation to r_t leads to:

$$R_t \approx 1 + \frac{1}{\gamma}\left(\frac{I_{t+1}}{I_t} - 1\right).$$

A similar though not identical approximation is given in Bettencourt and Ribeiro (2008), where $I_{t+1}/I_t - 1$ is replaced by $\log(I_{t+1}/I_t)$. Critically, incident infections only enter right-hand side above as a *ratio* of values adjacent in time, and thus if we are only able to estimate this up to an unknown multiplicative factor (due to confounding), then this factor approximately cancels in the ratio as long as it is slowly varying in time. In slightly more detail (and for simplicity, considering just a single location), suppose as before that a fraction a_t of infections go untested at time t. Then $I_t = b_t x_t$ where x_t is the number of new infections at time t that show up in case reports (i.e., the focus of this

paper) and $b_t = 1/(1-a_t)$. From the previous display,

$$R_t \approx 1 + \frac{1}{\gamma}\left(\frac{b_{t+1}x_{t+1}}{b_t x_t} - 1\right) \approx 1 + \frac{1}{\gamma}\left(\frac{x_{t+1}}{x_t} - 1\right),$$

where the last approximation is motivated by an additional assumption the untested fraction varies slowly over time (so $b_{t+1}/b_t \approx 1$). This shows that estimates of x_t can produce *approximately unconfounded* estimates of R_t, even though x_t is itself confounded due to a lack of universal testing. This is true both in the retrospective and real-time sense, and will be the topic of future study.

Evaluation via Reconvolution. An important avenue for evaluating our methodology (beyond evaluating against finalized infection rate estimates, as we do in this paper), would be to reconvolve our real-time nowcasts of infection rates forward in time in order to predict future case rates, and evaluate these predictions against finalized case reporting data. Making and evaluating point predictions would be relatively straightforward, however, distributional forecasts are currently the standard in epidemiological forecasting (and also in COVID-19 forecasting), and adding a distributional layer to our nowcasts (and propagating this through the convolution operator) requires substantial new developments, and we leave it to future work.

ACKNOWLEDGMENTS

APPENDIX A: ADMM FOR SOLVING DECONVOLUTION PROBLEMS

Here we give details on the ADMM approach used to solve the regularized least squares deconvolution problems in Sections 3 and 4. We first focus on problem (5), and then we discuss the modifications needed when incorporating extra regularization for real-time deconvolution as in (10). To simplify notation, we will henceforth drop the subscript dependnece of all quantities on the location ℓ, as well as the superscript dependence on the nowcast date t for the real-time problems.

We also use $\hat{P}$ to denote the (Toeplitz) convolution matrix with rows determined by $\hat{p}_s$, $s < t$, i.e., such that for any vector x (of appropriate dimension)

$$(\hat{P}x)_s = \sum_{k=1}^{d} \hat{p}_k x_{s-k}.$$

(We leave the dimensions of $\hat{P}$ and x here purposely ambiguous, which should always be clear from the context anyway; this allows us to borrow similar notation across problems with different underlying dimensions.) Thus we can rewrite (5) as

$$\underset{x}{\text{minimize}} \ \|y - \hat{P}x\|_2^2 + \lambda\|D^{(4)}x\|_1.$$

To apply ADMM, we must introduce auxiliary variables, and as in Ramdas and Tibshirani (2016), we use the following "specialized" decomposition (which improves the convergence speed):

$$\underset{x}{\text{minimize}} \ \|y - \hat{P}x\|_2^2 + \lambda\|D^{(1)}\alpha\|_1$$

$$\text{subject to } \alpha = D^{(3)}x,$$

where we used the recursive nature of the difference operators, writing the 4th-order operator as a product of the 1st- and 3rd-order operators: $D^{(4)} = D^{(1)}D^{(3)}$. The above problem gives rise to the augmented Lagrangian:

$$\mathcal{L}(x, \alpha, u) = \|y - \hat{P}x\|_2^2 + \lambda\|D^{(1)}\alpha\|_1 +$$

$$\rho\|\alpha - D^{(3)}x + u\|_2^2 - \rho\|u\|_2^2,$$

which corresponds to following ADMM updates, writing $D = D^{(3)}$ for brevity:

$$x \leftarrow (\hat{P}^T\hat{P} + \rho D^T D)^{-1}(\hat{P}^T y + \rho D^T(\alpha + u))$$

$$\alpha \leftarrow \underset{z}{\text{argmin}} \ \|Dx - u - z\|_2^2 + \frac{\lambda}{\rho}\|D^{(1)}\alpha\|_1$$

$$u \leftarrow u + \alpha - Dx.$$

The α-update here requires solving a 1-dimensional fused lasso problem, which can be done in linear-time with the dynamic programming approach of Johnson (2013). The x-update is more expensive than in pure trend filtering (with no convolution operator) but owing to the bandedness of $\hat{P}$ (and D, though the bandwidth d of $\hat{P}$ dominates), it can still be solved in $O(nd)$ operations. Further, in this and all applications of ADMM, we follow the recommendation of Ramdas and Tibshirani (2016) and set the Lagrangian parameter equal to the tuning parameter, $\rho = \lambda$.

As for the two extensions presented in (10), the natural trend filtering constraints can be be enforced by introducing a linear interpolant matrix as described in Section 11.2 of Tibshirani (2020). This effectively replaces the convolution matrix $\hat{P}$ and the 3rd difference operator D, in the ADMM steps above, by $\tilde{P}$ and $\tilde{D}$, respectively, which are given by right multiplying P and D by the interpolant matrix.

Moreover, the additional tapered smoothing term can be pushed into the augmented Lagrangian, and only alters the x-update, now becoming:

$$x \leftarrow (\tilde{P}^T\tilde{P} + \gamma M^T M + \rho\tilde{D})^{-1} \cdot$$

$$(\tilde{P}^T y + \rho\tilde{D}^T(\alpha + u)),$$

where M is the matrix $W^{(t)}D^{(1)}$ in the tapered penalty in (10) times the linear interpolant matrix.

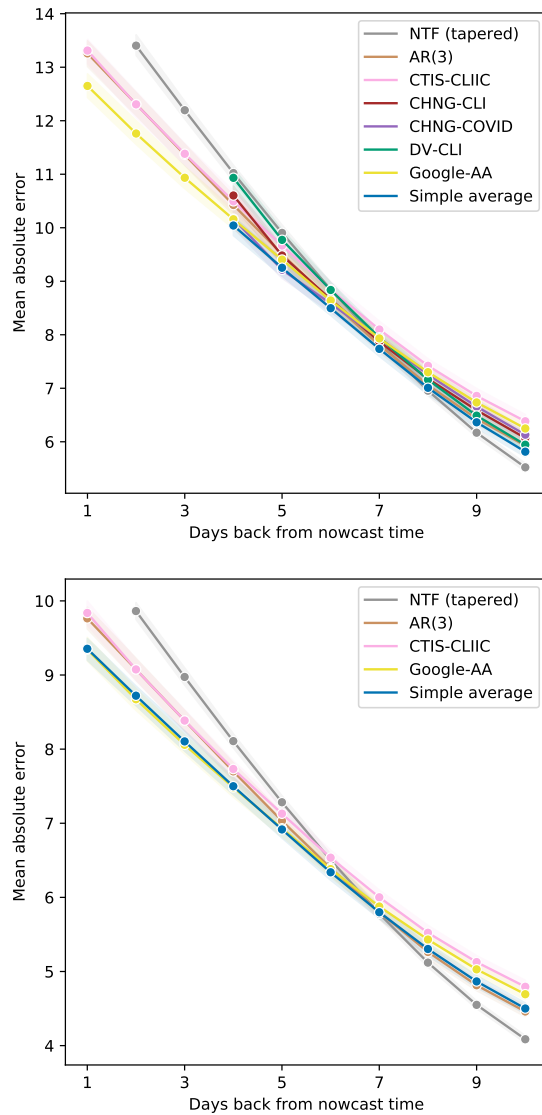FIG 15. *As in Figure 11, but including Google-AA.*

FIG 16. *As in Figure 12, but including Google-AA.*

APPENDIX B: ADDITIONAL EVALUATION RESULTS

Figures 15 and 16 are analogous to Figures 11 and 12, but with the inclusion of the Google-AA sensor. Similarly, Figures 17 and 18 are the counterparts to Figures 13 and 14, but with the inclusion of claims-based sensors.

REFERENCES

ABBOTT, S., HELLEWELL, J., THOMPSON, R. N., SHERRATT, K., GIBBS, H. P., BOSSE, N. I., MUNDAY, J. D., MEAKIN, S., DOUGHTY, E. L., CHUN, J. Y., CHAN, Y.-W. D., FINGER, F., CAMPBELL, P., ENDO, A., PEARSON, C. A. B., GIMMA, A., RUSSELL, T., CMMID COVID MODELLING GROUP, FLASCHE, S., KUCHARSKI, A. J., EGGO, R. M. and FUNK, S. (2020). Estimating the time-varying reproduction number of SARS-CoV-2 using national and subnational case counts. *Wellcome Open Research* **5**.

ACKLEY, A. F., PILEWSKI, S., PETROVIC, V. S., WORDEN, L., MURRAY, E. and PORCO, T. C. (2020). assessing the utility of a smart thermometer and mobile application as a surveillance tool for in-
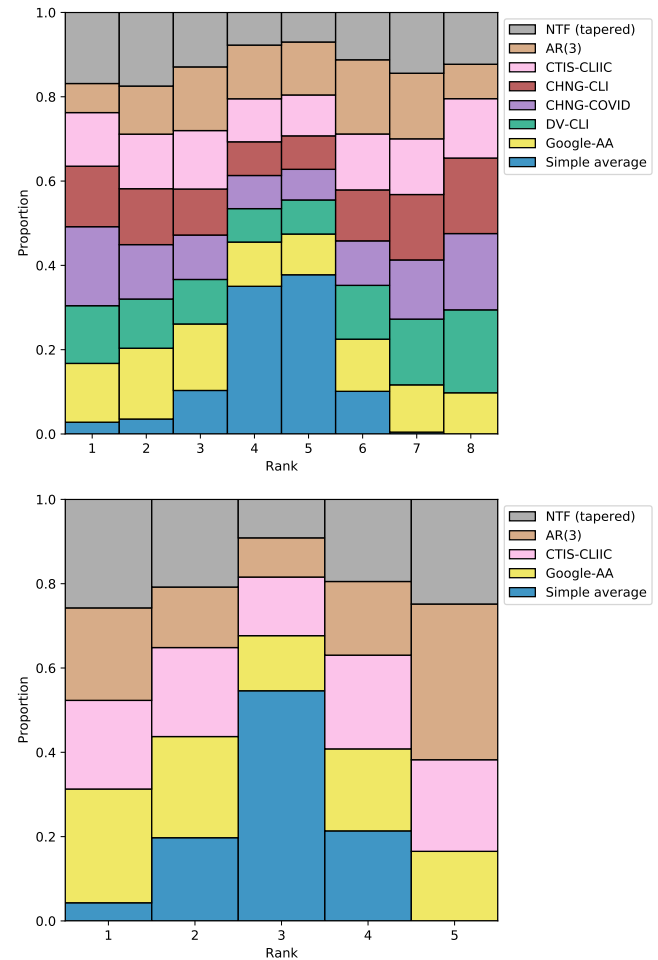
fluenza and influenza-like illness. *Health Informatics Journal* **26** 2148–2158.

BAVADEKAR, S., DAI, A., DAVIS, J., DESFONTAINES, D., ECKSTEIN, I., EVERETT, K., FABRIKANT, A., FLORES, G., GABRILOVICH, E., GADEPALLI, K., GLASS, S., HUANG, R., KAMATH, C., KRAFT, D., KUMOK, A., MARFATIA, H., MAYER, Y., MILLER, B., PEARCE, A., PERERA, I. M., RAMACHANDRAN, V., RAMAN, K., ROESSLER, T., SHAFRAN, I., SHEKEL, T., STANTON, C., STIMES, J., SUN, M., WELLENIUS, G., and ZOGHI, M. (2020). Google COVID-19 search trends symptoms dataset: Anonymization process description. arXiv: 2009.01265.

BETTENCOURT, L. M. A. and RIBEIRO, R. M. (2008). Real time Bayesian estimation of the epidemic potential of emerging infectious diseases. *PLOS ONE* **3** e2185.

BROOKS, L. C. (2020). Pancasting: Forecasting epidemics from provisional data, PhD thesis, Carnegie Mellon University.

BROWNSTEIN, J. S., FREIFELD, C. C. and MADOFF, L. C. (2009). Digital disease detection — harnessing the web for public health surveillance. *New England Journal of Medicine* **360** 2153–2157.

CARLSON, S. J., DALTON, C. B., BUTLER, M. T., FEJSA, J., ELVIDGE, E. and DURRHEIM, D. N. (2013). Flutracking weekly online community survey of influenza-like illness annual report 2011 and 2012. *Communicable diseases intelligence quarterly report* **37** E398–406.

CHARU, V., ZEGER, S., GOG, J., BJØRNSTAD, O. N., KISSLER, S., SIMONSEN, L., GRENFELL, B. T. and VIBOUD, C. (2017). Human
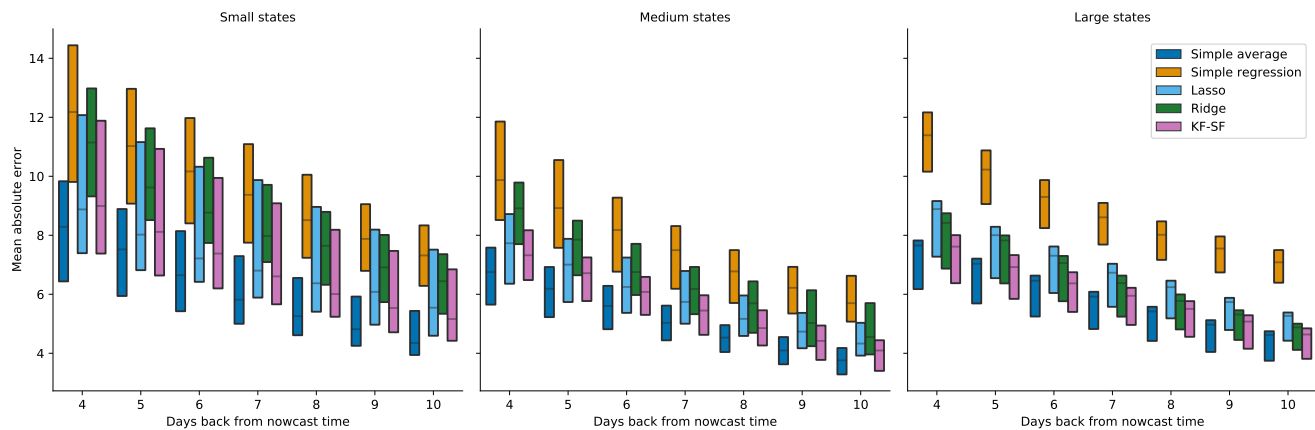
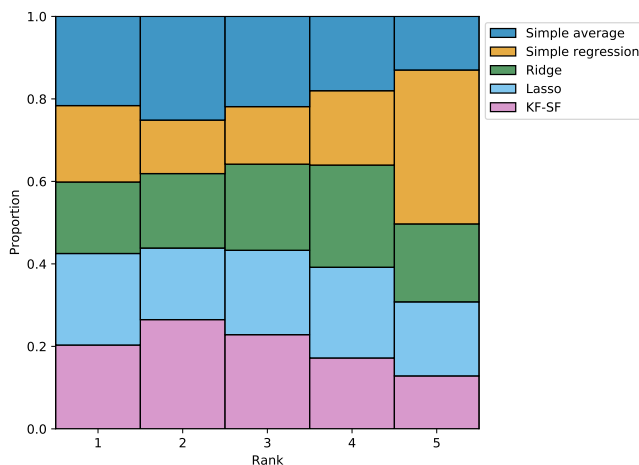FIG 17. *As in Figure 13, but including claims-based signals.*

FIG 18. *As in Figure 14, but including claims-based signals.*

mobility and the spatial transmission of influenza in the United States. *PLOS Computational Biology* **13** 1–23.

CHITWOOD, M. H., RUSSI, M., GUNASEKERA, K., HAVUMAKI, J., PITZER, V. E., SALOMON, J. A., SWARTWOOD, N., WARREN, J. L., WEINBERGER, D. M. and COHEN, T. (2021). Reconstructing the course of the COVID-19 epidemic over 2020 for US states and counties: results of a Bayesian evidence synthesis model.

CORI, A., FERGUSON, N. M., FRASER, C. and CAUCHEMEZ, S. (2013). A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology* **178** 1505–1512.

DEBEYE, H. W. J. and VAN RIEL, P. (1990). L_p-norm deconvolution. *Geophysical Prospecting* **38** 381–403.

DONG, E., DU, H. and GARDNER, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases* **20** 533–544.

FARROW, D. C. (2016). Modeling the past, present, and future of influenza, PhD thesis, Carnegie Mellon University.

CENTERS FOR DISEASE CONTROL AND PREVENTION, COVID-19 RESPONSE (2020a). COVID-19 Case Surveillance Public Use Data. https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/vbim-akqf. Data accessed on November 3, 2021.

CENTERS FOR DISEASE CONTROL AND PREVENTION, COVID-19 RESPONSE (2020b). COVID-19 Case Surveillance Restricted Access Detailed Data. https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Restricted-Access-Detai/mbd7-r32t. Data accessed on November 3, 2021.

GINSBERG, J., MOHEBBI, M. H., PATEL, R. S., BRAMMER, L., SMOLINSKI, M. S. and BRILLIANT, L. (2009). Detecting influenza epidemics using search engine query data. *Nature* **457** 1012–1014.

GOLDSTEIN, E., DUSHOFF, J., MA, J., PLOTKIN, J. B., EARN, D. J. and LIPSITCH, M. (2009). Reconstructing influenza incidence by deconvolution of daily mortality time series. *Proceedings of the National Academy of Sciences* **106** 21825–21829.

GOSTIC, K. M., MCGOUGH, L., BASKERVILLE, E. B., ABBOTT, S., JOSHI, K., TEDIJANTO, C., KAHN, R., NIEHUS, R., HAY, J. A., DE SALAZAR, P. M., HELLEWELL, J., MEAKIN, S., MUNDAY, J. D., BOSSE, N. I., SHERRAT, K., THOMPSON, R. N., WHITE, L. F., HUISMAN, J. S., SCIRE, J., BONHOEFFER, S., STADLER, T., WALLINGA, J., FUNK, S., LIPSITCH, M. and COBEY, S. (2020). Practical considerations for measuring the effective reproductive number, R_t. *PLOS Computational Biology* **16** 1–21.

HAWRYLUK, I., HOELTGEBAUM, H., MISHRA, S., MISCOURIDOU, X., SCHNEKENBERG, R. P., WHITTAKER, C., VOLLMER, M., FLAXMAN, S., BHATT, S. and MELLAN, T. A. (2021). Gaussian process nowcasting: Application to COVID-19 mortality reporting. In *Conference on Uncertainty in Artificial Intelligence*.

JAHJA, M., FARROW, D., ROSENFELD, R. and TIBSHIRANI, R. J. (2019). Kalman Filter, sensor fusion, and constrained regression: equivalences and insights. In *Advances in Neural Information Processing Systems*.

JOHNSON, N. (2013). A dynamic programming algorithm for the fused lasso and L_0-segmentation. *Journal of Computational and Graphical Statistics* **22** 246–260.

KAPLAN, E. L. and MEIER, P. (1958). Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association* **53** 457–481.

KASS-HOUT, T. A. and ALHINNAWI, H. (2013). Social media in public health. *British Medical Bulletin* **108** 5–24.

KASS-HOUT, T. A. and ZHANG, X. (2011). *Biosurveillance: Methods and Case Studies*. CRC Press.

REICH LAB (2020). The COVID-19 Forecast Hub. https://covid19forecasthub.org.

LEUBA, S. I., YAESOUBI, R., ANTILLON, M., COHEN, T. and ZIMMER, C. (2020). Tracking and predicting U.S. influenza activity with a real-time surveillance network. *PLOS Computational Biology* **16** 1–14.

MCDONALD, D. J., BIEN, J., GREEN, A., HU, A. J., DEFRIES, N., HYUN, S., OLIVEIRA, N. L., SHARPNACK, J., TANG, J., TIBSHIRANI, R., VENTURA, V., WASSERMAN, L. and TIBSHIRANI, R. J. (2021). Can auxiliary indicators improve COVID-19 forecasting and hotspot prediction? e2111453118. To appear, PNAS.

MCGOUGH, S. F., JOHANSSON, M. A., LIPSITCH, M. and MENZIES, N. A. (2020). Nowcasting by Bayesian smoothing: A flexible, generalizable model for real-time epidemic tracking. *PLOS Computational Biology* **16** e1007735.

MCIVER, D. J. and BROWNSTEIN, J. S. (2014). Wikipedia usage estimates prevalence of influenza-like illness in the United States in near real-time. *PLOS Computational Biology* **10** e1003581.

OPPENHEIM, A. V. and VERGHESE, G. C. (2017). *Signals, Systems and Inference*. Pearson.

PAUL, M. J. and DREDZE, M. (2017). Social monitoring for public health. *Synthesis Lectures on Information Concepts, Retrieval, and Services* **9** 1–183.

RADIN, J. M., WINEINGER, N. E., TOPOL, E. J. and STEINHUBL, S. R. (2020). Harnessing wearable device data to improve state-level real-time surveillance of influenza-like illness in the USA: A population-based study. *The Lancet Digital Health* **2** e85–e93.

RAMDAS, A. and TIBSHIRANI, R. J. (2016). Fast and Flexible ADMM Algorithms for Trend Filtering. *Journal of Computational and Graphical Statistics* **25** 839–858.

REINHART, A., BROOKS, L., JAHJA, M., RUMACK, A., TANG, J., AGRAWAL, S., SAEED, W. A., ARNOLD, T., BASU, A., BIEN, J., CABRERA, Á. A., CHIN, A., CHUA, E. J., CLARK, B., COLQUHOUN, S., DEFRIES, N., FARROW, D. C., FORLIZZI, J., GRABMAN, J., GRATZL, S., GREEN, A., HAFF, G., HAN, R., HARWOOD, K., HU, A. J., HYDE, R., HYUN, S., JOSHI, A., KIM, J., KUZNETSOV, A., MOTTE-KERR, W. L., LEE, Y. J., LEE, K., LIPTON, Z. C., LIU, M. X., MACKEY, L., MAZAITIS, K., MCDONALD, D. J., MCGUINNESS, P., NARASIMHAN, B., O'BRIEN, M. P., OLIVEIRA, N. L., PATIL, P., PERER, A., POLITSCH, C. A., RAJANALA, S., RUCKER, D., SCOTT, C., SHAH, N. H., SHANKAR, V., SHARPNACK, J., SHEMETOV, D., SIMON, N., SMITH, B. Y., SRIVASTAVA, V., TAN, S., TIBSHIRANI, R., TUZHILINA, E., NORTWICK, A. K. V., VENTURA, V., WASSERMAN, L., WEAVER, B., WEISS, J. C., WHITMAN, S., WILLIAMS, K., ROSENFELD, R. and TIBSHIRANI, R. J. (2021). An open repository of real-time COVID-19 indicators. *Proceedings of the National Academy of Sciences* **51** e2111452118.

ROSENFELD, R. and TIBSHIRANI, R. J. (2021). Epidemic tracking and forecasting: Lessons learned from a tumultuous year. *Proceedings of the National Academy of Sciences* **51** e2111456118.

RUDIN, L. I. and OSHER, S. (1994). Total variation based image restoration with free local constraints. In *International Conference on Image Processing* **1** 31–35.

SALATHÉ, M., BENGTSSON, L., BODNAR, T. J., BREWER, D. D., BROWNSTEIN, J. S., BUCKEE, C., CAMPBELL, E. M., CATTUTO, C., KHANDELWAL, S., MABRY, P. L. and VESPIGNANI, A. (2012). Digital epidemiology. *PLOS Computational Biology* **8** 1–3.

SALOMON, J. A., REINHART, A., BILINSKI, A., CHUA, E. J., LA MOTTE-KERR, W., RÖNN, M. M., REITSMA, M., MORRIS, K. A., LaROCCA, S., FARAG, T., KREUTER, F., ROSENFELD, R. and TIBSHIRANI, R. J. (2021). The COVID-19 trends and impact survey: Continuous real-time measurement of COVID-19 symptoms, risks, protective behaviors, testing and vaccination. *Proceedings of the National Academy of Sciences* **51** e2111454118.

SANTILLANA, M., NGUYEN, A. T., DREDZE, M., PAUL, M. J., NSOESIE, E. O. and BROWNSTEIN, J. S. (2015). Combining search, social media, and traditional data sources to improve influenza surveillance. *PLOS Computational Biology* **11** e1004513.

SANTILLANA, M., NGUYEN, A. T., LOUIE, T., ZINK, A., GRAY, J., SUNG, I. and BROWNSTEIN, J. S. (2016). Cloud-based electronic health records for real-time, region-specific influenza surveillance. *Scientific Reports* **6** 1–8.

SMOLINSKI, M. S., CRAWLEY, A. W., BALTRUSAITIS, K., CHUNARA, R., OLSEN, J. M., WÓJCIK, O., SANTILLANA, M., NGUYEN, A. and BROWNSTEIN, J. S. (2015). Flu Near You: Crowdsourced symptom reporting spanning 2 influenza seasons. *American Journal of Public Health* **105** 2124–2130.

SYSTROM, K., VLADEK, T. and KRIEGER, M. (2020). Rt.live. https://github.com/rtcovidlive/covid-model.

TAYLOR, H. L., BANKS, S. C. and MCCOY, J. F. (1979). Deconvolution with the ℓ_1 norm. *Geophysics* **44** 39–52.

THOMPSON, R. N., STOCKWIN, J. E., VAN GAALEN, R. D., POLONSKY, J. A., KAMVAR, Z. N., DEMARSH, P. A., DAHLQWIST, E., LI, S., MIGUEL, E., JOMBART, T., LESSLER, J., CAUCHEMEZ, S. and CORI, A. (2019). Improved inference of time-varying reproduction numbers during infectious disease outbreaks. *Epidemics* **29** 100356.

TIBSHIRANI, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics* **42** 285–323.

TIBSHIRANI, R. J. (2020). Divided differences, falling factorials, and discrete splines: Another look at trend filtering and related problems. arXiv: 2003.03886.

VIBOUD, C., CHARU, V., OLSON, D., BALLESTEROS, S., GOG, J., KHAN, F., GRENFELL, B. and SIMONSEN, L. (2014). Demonstrating the use of high-volume electronic medical claims data to monitor local and regional influenza activity in the US. *PLOS ONE* **9** 1–12.

WALLINGA, J. and LIPSITCH, M. (2007). How generation intervals shape the relationship between growth rates and reproductive numbers. *Proceedings of the Royal Society B: Biological Sciences* **274** 599–604.

WIENER, N. (1964). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press.

YANG, S., SANTILLANA, M. and KOU, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences* **112** 14473–14478.

YANG, C.-Y., CHEN, R.-J., CHOU, W.-L., LEE, Y.-J. and LO, Y.-S. (2019). An integrated influenza surveillance framework based on national influenza-like illness incidence and multiple hospital electronic medical records for early prediction of influenza epidemics: Design and evaluation. *Journal of Medical Internet Research* **21** e12341.