Electronic Companion to "Small Area Estimation of Case Growths for Timely COVID-19 Outbreak Detection"

The appendix provides additional implementation details for the methods discussed in the main text, supplementary benchmarking results, and further analyses conducted on the case study. It begins with Appendix A, which details the implementation of `TLRF`, compares it with alternative approaches, and analyzes the robustness of our chosen method. Appendix B analyzes the limitations of the generalized random forests (GRF) by Athey et al. (2019) in our context. Appendix C follows with additional proofs and analytical results for the lemmas and theorems introduced in the main text. D presents the details of our synthetic experiment to assess the robustness of `TLRF` against model misspecification. In Appendix E, we outline the implementation details of our benchmarks. Appendix F provides information on the datasets used in our study and includes an analysis of the most important features. Appendix G expands on the case study with additional policies and use cases for the `TLRF` estimator. Finally, Appendix H discusses the COVID-19 Outbreak Detection Tool, an online decision support tool made available to the public.

Appendix A: Further Details Regarding TLRF: Fine-tuning and Analyses

This section provides additional details on the `TLRF` algorithm and investigation into other potential choices of implementation. In Appendix A.1, we present the pseudo-code needed to implement `TLRF`. Then, in Appendix A.2, we present, derive, and compare alternative TLRF-based implementations. Finally, Appendix A.3 examines the robustness of `TLRF` by analyzing how properties of the underlying estimator, such as leaf sizes and maximum tree depth, vary with different hyperparameter choices.

A.1. Transfer Learning Random Forests (`TLRF`) Pseudo-code

This subsection presents the pseudo-code required to implement `TLRF`. The process begins with the normalization and initial estimation step, outlined in Algorithm 1. Following this, the transfer learning step is performed using a standard regression random forest algorithm `regression_forest()` (Algorithm 2).

Algorithm 1 Normalization and Initial Estimates

Input: $\{(I_{t',c'}, \vec{X}_{t',c'})\}_{t' \in \{1,\ldots,t\},\, c \in C}$

Output: $\{Feature[t',c']\}_{t' \in \{1,\ldots,t\},\, c' \in C}$, $\{Y[t',c',1], Y[t',c',0]\}_{t' \in \{1,\ldots,t\},\, c' \in C}$

for $c' = 1, \ldots, |C|$ **do**
  for $t' = 1, \ldots, t$ **do**
    Normalize the incident case numbers for each block
    $Y[t',c',1] \leftarrow \ln(I_{t',c'}) - \ln(I_{t'-1,c'})$
    $Y[t',c',0] \leftarrow 0$
    $W[t',c',1] \leftarrow 1$
    $W[t',c',0] \leftarrow 0$
    Generate the initial OLS estimates for each block
    $Dep \leftarrow \{Y[t',c',1], Y[t',c',0]\}$
    $Ind \leftarrow \{W[t',c',1], W[t',c',0]\}$
    $(r^{\text{ols}}_{t',c'}, \alpha^{\text{ols}}_{t',c'}) \leftarrow OLS(Dep, Ind)$
    Append the feature data for each block
    $Feature[t',c'] \leftarrow \{\vec{X}_{t',c'}, r^{\text{ols}}_{t',c'}, \alpha^{\text{ols}}_{t',c'}, t\}$
  end for
end for

Algorithm 2 `TLRF` through the `regression_forest`() implementation

Input: $\{Feature[t',c']\}_{t' \in \{1,\ldots,t\},\, c' \in C}$, $\{Y[t',c',1], Y[t',c',0]\}_{t' \in \{1,\ldots,t\},\, c' \in C}$

Output: $\{r^{TLRF}_{t,c'}\}_{c' \in C}$

Compute the congruence classes
$[t] \leftarrow \{t' \in \{1,\ldots,t\} \mid t \equiv t' \pmod 2\}$
Assign the feature variable
$\vec{X} \leftarrow \{Feature[t',c']\}_{t' \in [t],\, c' \in C}$
Assign the outcome variable
$Y \leftarrow \{Y[t',c',1]\}_{t' \in [t],\, c' \in C}$
Run the random forest and obtain final estimates
$\{r^{TLRF}_{t,c'}\}_{c' \in C} \leftarrow$ `regression_forest`$(\vec{X}, Y)$

A.2. Other TLRF Based Implementations

This subsection provides supplementary results for alternative implementation choices within the TLRF framework. We present analytical derivations, implementation details, and empirical comparisons against TLRF. First, in Appendix A.2.1, we investigate whether TLRF's improvement over fixed and dynamic time-window methods stems solely from its advanced time-based cross-validation techniques or also from its integration of spatial information. To test this, we benchmark against the `TLRF Time Only` estimator, which excludes spatial information, and show that spatial information is critical for improved performance. Second, in Appendix A.2.2, we examine the impact of varying the initial estimator's fitting window size (δ), where TLRF uses the minimal size of $\delta = 2$. We derive alternative estimators for other larger δ values, referred to as TLRF δ, and empirically demonstrate that $\delta = 2$ offers the best performance.

A.2.1. TLRF vs TLRF Time Only Comparison We investigate whether the improvement of TLRF over fixed/dynamic time window methods is attributable solely to its advanced time-based cross-validation techniques or to its additional integration of spatial information. To that end, we consider another potential choice of implementation, `TLRF Time Only`, that restricts TLRF's access to spatial information. By comparing this new benchmark with TLRF and the existing methods, we find that incorporating spatial information is essential for TLRF's superior performance to fixed/dynamic time window methods.

We now describe the `TLRF Time Only` benchmark. This benchmark removes access to spatial information by having a TLRF trained separately for each county. That is, each county's incidence case history and time-varying features are fed separately into a TLRF training process, generating a unique estimator for every county every day. With over 1,000 days and 3,000 counties in our study sample, this experiment results in over 3 million separately trained TLRF estimators in total. We discuss and analyze its performance below.

Method	MAE	RMSE
`TLRF Time Only`	0.216	0.291
TLRF	**0.126**	**0.195**

Table 1 Median MAE and RMSE of TLRF vs. `TLRF Time Only`

As indicated in Table 1, the performance superiority of TLRF over fixed/dynamic time window methods is not replicated by `TLRF Time Only`. Specifically, when the access to spatial information is restricted, TLRF

`Time Only` is outperformed by both dynamic window approaches (i.e., `tcv` and `ctcv`) in terms of median RMSE. In contrast, with access to spatial information, `TLRF` outperforms the dynamic window approaches in both metrics. Therefore, the improvement of `TLRF` over the fixed/dynamic window approaches is not solely due to smarter pooling over time, but from its access to spatial information.

To further investigate `TLRF Time Only`'s decrease in performance with respect to `TLRF`, we decided to inspect the mean and median depths of each tree in both `TLRF` and `TLRF Time Only`.

As seen in Figure 1, `TLRF Time Only`'s trees are considerably more shallow than that of `TLRF`'s, where the latter's depths are typically 10 times that of the former. It is clear that the historical data of one county, which is what `TLRF Time Only` has access to, contains significantly less information than that of the entire country, which is utilized by `TLRF`. This comparison shows that the lack of access to spatial information hinder `TLRF Time Only`'s ability to properly conduct transfer learning, resulting in decreased performance.

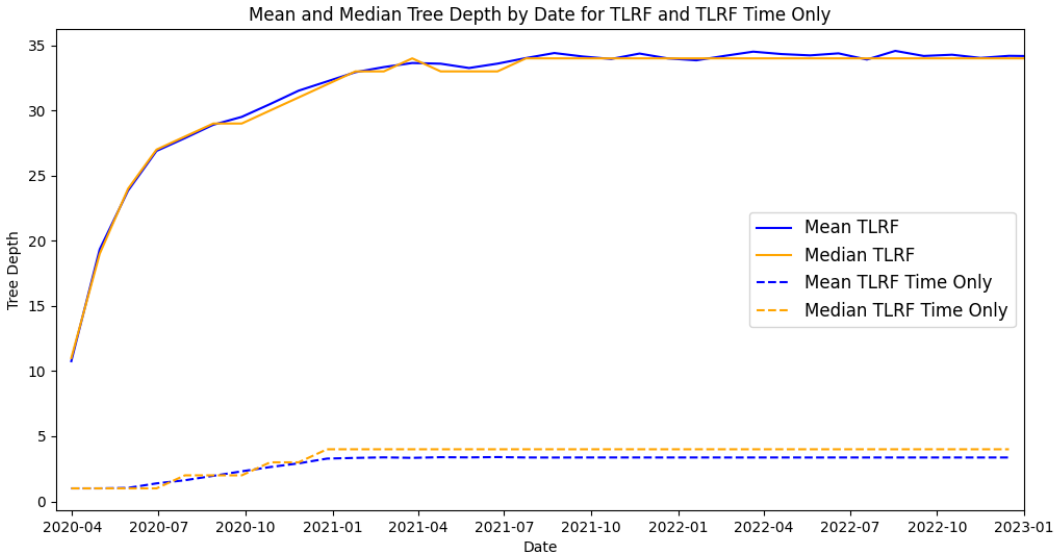

Figure 1 Mean and median tree depth by date (`TLRF` vs. `TLRF Time Only`)

A.2.2. TLRF δ Estimators To derive the `TLRF` δ estimators (6), this section follows the same steps as in §4. Specifically, we first characterize the conditional mean outcome corresponding to the initial estimator and subsequently the weighted conditional mean outcome that yields the `TLRF` δ estimators.

First, Lemma 1 characterizes the conditional mean outcome of an initial estimator with a δ-day fitting window.

LEMMA 1. *Under OLS Assumptions 1 and 2, an instantaneous county-level exponential growth rate $r_{t,c}$ with realized features $\vec{X}_{t,c}$ can be estimated from data $\{\ln(I_{t^-,c}), \ln(I_{t^--1,c}), \vec{X}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ using the OLS*

estimator with a δ-day fitting window (25). The corresponding conditional mean outcome of this estimator is

$$r(\vec{X}_{t,c}) = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c}\mathbb{E}[(\ln(\boldsymbol{I_{t,c}}) - \ln(\boldsymbol{I_{t-1,c}}))|\vec{X}_{t,c}], \tag{1}$$

where the weights $\{\hat{\mu}_{t^-,c}\}_{t^-\in\{t-\delta+2,\ldots,t\}}$ are defined in (24).

Proof of Lemma 1:

By Theorems 2 and 3, we can rewrite the conditional mean outcome corresponding to the OLS estimator with a δ-day fitting window (23) as (1). □

Second, Theorem 1 characterizes the weighted conditional mean outcomes that yield the TLRF δ estimators.

THEOREM 1. *For any county $c, c' \in C$ on any day $t' \in [t^-]$ with realized features $\vec{X}_{t^-,c}$ and $\vec{X}_{t',c'}$ such that $\vec{X}_{t',c'} \in \bigcup_{b\in B} L_b(\vec{X}_{t^-,c})$, its instantaneous county-level exponential growth rate $r_{t,c}$ can be estimated from data $\{\{\ln(I_{t',c'}), \ln(I_{t'-1,c'}), \vec{X}_{t',c'}\}_{t'\in[t^-],c'\in C}\}_{t^-\in\{t-\delta+2,\ldots,t\}}$ via the weighted conditional outcomes:*

$$r(\vec{X}_{t,c}) = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \sum_{c'\in C} \sum_{t'\in[t^-]} \gamma_{t',c'}(\vec{X}_{t^-,c})\mathbb{E}[(\ln(\boldsymbol{I_{t',c'}}) - \ln(\boldsymbol{I_{t'-1,c'}}))|\vec{X}_{t^-,c}], \tag{2}$$

resulting in the following TLRF δ estimator:

$$\hat{\boldsymbol{r}}^{\boldsymbol{TLRF}(\delta)}_{\boldsymbol{t,c}} = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \sum_{c'\in C} \sum_{t'\in[t^-]} \boldsymbol{\gamma_{t',c'}}(\vec{X}_{t^-,c}) \left(\ln(\boldsymbol{I_{t',c'}}) - \ln(\boldsymbol{I_{t'-1,c'}})\right), \tag{3}$$

where weights $\{\hat{\mu}_{t^-,c}\}_{t^-\in\{t-\delta+2,\ldots,t\}}$ are defined in (24).

Proof of Theorem 1:

By Theorem 2, $\forall t^- \in \{t-\delta+2,\ldots,t\}$, $r_{t^-,c}$ can be estimated from data $\{\ln(I_{t',c'}), \ln(I_{t'-1,c'}), \vec{X}_{t',c'}\}_{t'\in[t^-],c'\in C}$ via the weighted conditional mean outcome:

$$r(\vec{X}_{t^-,c}) = \sum_{c'\in C} \sum_{t'\in[t^-]} \gamma_{t',c'}(\vec{X}_{t^-,c})\mathbb{E}[(\ln(\boldsymbol{I_{t',c'}}) - \ln(\boldsymbol{I_{t'-1,c'}}))|\vec{X}_{t^-,c}]. \tag{4}$$

In addition, (4) implies (2) by Theorem 3. Lastly, (3) is the sample counterpart of (2). □

Performance Comparison Against TLRF δ

We can thus extend Theorem 1 and Theorem 2 by allowing initial estimators to have fitting window sizes exceeding $\delta = 2$. This extension essentially merges OLS weights with TLRF weighting schemes. Specifically, under the additional assumptions specified in Theorem 1, the initial estimator of TLRF δ becomes an OLS estimator with a δ-day fitting window, i.e.,

$$\hat{\boldsymbol{r}}^{\boldsymbol{ols}(\delta)}_{\boldsymbol{t,c}} = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I_{t^-,c}}) - \ln(\boldsymbol{I_{t^-1,c}})\right), \tag{5}$$

which coincides with the TLRF initial estimator (9) only when $\delta = 2$. Additionally, Theorem 1 demonstrates that these initial estimators yield the subsequent TLRF δ estimator, i.e., $\forall \delta \in \{2, \ldots, t\}$,

$$\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}(\delta)} = \sum_{t^- = t-\delta+2}^{t} \hat{\mu}_{t^-,c} \sum_{c' \in C} \sum_{t' \in [t^-]} \boldsymbol{\gamma}_{\boldsymbol{t}',\boldsymbol{c}'}(\vec{\boldsymbol{X}}_{\boldsymbol{t}^-,\boldsymbol{c}}) \left(\ln(\boldsymbol{I}_{\boldsymbol{t}',\boldsymbol{c}'}) - \ln(\boldsymbol{I}_{\boldsymbol{t}'-\boldsymbol{1},\boldsymbol{c}'}) \right). \tag{6}$$

where the non-adaptive weights $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ are specified in (7). Notably, the TLRF δ estimator (6) reduces to the TLRF estimator (12) only when $\delta = 2$.

As depicted in Table 2, the median RMSE and MAE values of the TLRF algorithm are presented for various selections of δ. Additionally, Figure 2 illustrates the daily mean MAE values corresponding to different choices of $\delta = 2$ vs $\delta = 4$ for clarity. The numerical findings provide evidence of TLRF's capacity to generalize effectively across diverse weighting schemes, with the results indicating that the most adaptable scheme, namely $\delta = 2$, exhibits the most favorable empirical performance.

Method	**MAE**	**RMSE**
TLRF 14	0.173	0.237
TLRF 7	0.153	0.226
TLRF 4	0.138	0.217
TLRF 3	0.132	0.203
TLRF (δ=2)	**0.126**	**0.195**

Table 2 Median MAE and RMSE of TLRF vs. TLRF δ: TLRF corresponds to TLRF δ with $\delta = 2$.

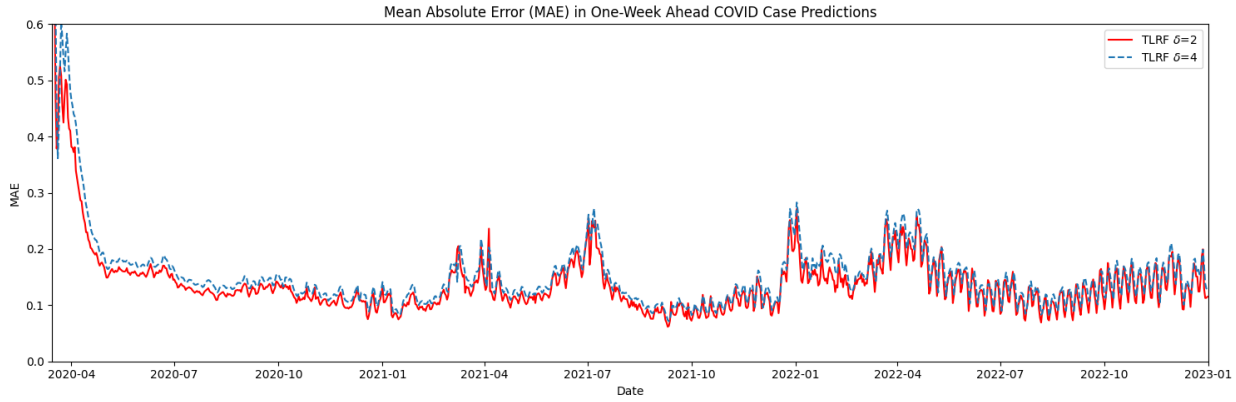

Figure 2 MAE plot of prediction accuracy (TLRF with choices $\delta = 2$ vs $\delta = 4$)

A.3. Robustness and Stability of the TLRF Algorithm

This subsection examines the robustness of the TLRF regarding its key hyperparameters and performance indicators, i.e., fitting window size for the initial estimator, number of trees, leaf node sizes, and tree depth.

Fitting Window Size for the Initial Estimator (δ)

Results of extensive computational experiments on the impact of δ on TLRF's performance are presented and discussed in Appendix A.2.2. These experiments reveal that the choice $\delta = 2$ achieves the best empirical performance.

Number of Trees

For the number of trees in TLRF, we originally followed the default setting (i.e., `num.trees` = 2000) in the `grf` package (c.f. Athey et al. 2019). To seek more efficient implementations of TLRF, we have conducted additional computational experiments and found that there was no substantial improvement in the algorithm's empirical performance beyond 200 trees. Therefore, in the weekly update of our online decision support tool, we decided to keep the number of trees at 200 to alleviate the computational burden.

Leaf Node Sizes

Our TLRF algorithm adopts the default leaf node size figuration (i.e., `min.node.size=5`) in the `grf` package (c.f. Athey et al. 2019). To investigate the potential impact of this hyperparameter on our algorithm's behavior, we have conducted additional computational experiments in this revision. Specifically, we examine whether this minimum node size hyperparameter forces the split terminations in TLRF. As illustrated in Figure 3, we have not found any evidence of this nature. Particularly, both the mean and median leaf node sizes observed in TLRF significantly surpass the default minimum node size, which suggests that this hyperparameter does not affect the quality of TLRF's splits. Therefore, the performance of TLRF is not affected by the default setting of minimum node size.

Tree Depth

Sufficient tree depth within the TLRF algorithm is important to effectively identify meaningful subpopulations for splitting. However, excessively deep trees may risk overfitting to specific subpopulations. This section focuses on the evolution of trees in TLRF with increasing training data volume and assesses whether an adequate tree depth is achieved by this algorithm. Figure 4 portrays a progression in tree depth of TLRF, initially averaging at a depth of 11 and eventually stabilizing around a depth of 33 as the dataset expanded. Additionally, Figure 3 reveals a consistency in both mean and median leaf node sizes of TLRF throughout our study period. Observations from Figures 3 and 4 highlight that the splitting behavior of TLRF remains consistent even with increasing data fed into the algorithm. Such evidence supports the notion that TLRF has achieved an adequate tree depth, allowing it to accurately classify large volumes of data.

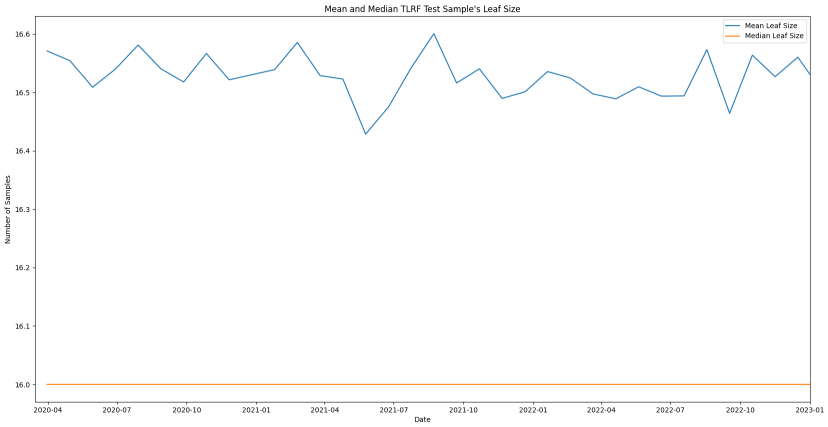

Figure 3 Mean and Median Leaf Sizes of TLRF

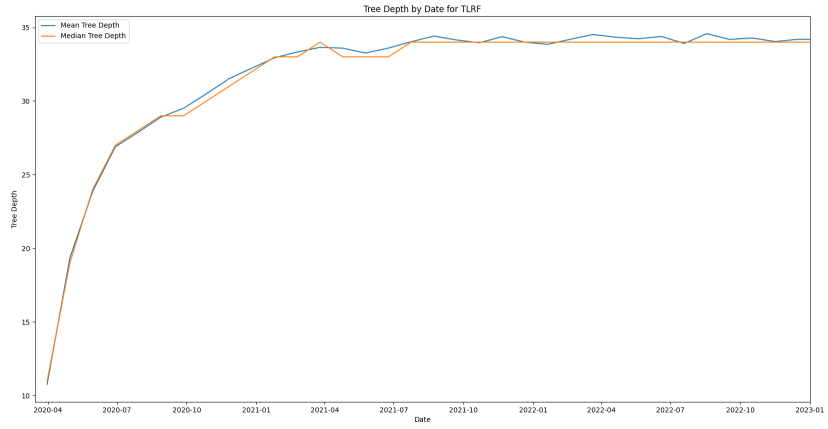

Figure 4 Mean and Median Tree Depth of TLRF

Appendix B: Estimation of $r_{t,c}$ through Direct GRF Implementation

This section estimates $r_{t,c}$ using an off-the-shelf GRF function `causal_forest()`. Specifically, to estimate the actual growth model (2), a typical input to the `causal_forest()` function is summarized by Table 3. The output of this function is the CAPE:

$$\frac{\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I_{s,c}}), \boldsymbol{s}|\vec{X}_{t,c}]}{\mathbb{V}\mathbb{a}\mathbb{r}[\boldsymbol{s}|\vec{X}_{t,c}]}, \tag{7}$$

representing the slope term estimate of the actual growth model (2). However, as we demonstrate both empirically and theoretically in the rest of this section, the estimate (7) only identifies $r_{t,c}$ when features remain constant over time: $\vec{X}_{t,c} = \vec{X}_c, \ \ \forall t \in \mathbb{Z}^+$.

Dependent Variable ($\ln(\boldsymbol{I}_{s,c})$)	Independent Variable (s)	Feature ($\vec{\boldsymbol{X}}_{s,c}$)
$\ln(I_{1,1})$	1	$\vec{X}_{1,1}$
$\ln(I_{2,1})$	2	$\vec{X}_{2,1}$
$\ln(I_{3,1})$	3	$\vec{X}_{3,1}$
$\ln(I_{4,1})$	4	$\vec{X}_{4,1}$
⋮	⋮	⋮

Table 3 Direct `causal_forest()` Implementation

First, we provide empirical evidence that a direct `causal_forest()` implementation summarized by Table 3 does not effectively utilize time-varying features. As illustrated in Figure 5 and Table 4, incorporating time-varying features into a direct `causal_forest()` implementation yields only a marginal enhancement in performance. Moreover, it remains ambiguous whether this improvement is attributable to the effective utilization of temporal information. This ambiguity arises because all features employed in this study, whether time-invariant or time-varying, inherently contain spatial information. For example, the county-level vaccination coverage rate is a time-varying feature, yet its dynamics are significantly shaped by state-specific policies, which reveal geographic information. Consequently, the performance improvement observed when time-varying features such as vaccination coverage rate are included could also be ascribed to the spatial information embedded within these features.

Method	**MAE**	**RMSE**
Direct `causal_forest()` Implementation without time varying features	0.218	0.291
Direct `causal_forest()` Implementation with time varying features	0.204	0.275
TLRF	**0.126**	**0.195**

Table 4 Median MAE and RMSE of TLRF vs. Direct `causal_forest()` Implementation with/without Time-varying Features

Second, we theoretically reveal why the direct `causal_forest()` implementation faces these limitations. Specifically, we note that

PROPOSITION 1. *The CAPE (7) can be decomposed as*

$$\frac{\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I}_{s,c}), s|\vec{\boldsymbol{X}} = \vec{X}_{t,c}]}{\mathbb{V}\mathbb{a}\mathbb{r}[s|\vec{\boldsymbol{X}} = \vec{X}_{t,c}]} = \sum_{t^{-}=2}^{t} \omega_{t^{-},c} \left(\mathbb{E}[\ln(\boldsymbol{I}_{s,c})|s = t^{-}, \vec{\boldsymbol{X}} = \vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}_{s,c})|s = t^{-} - 1, \vec{\boldsymbol{X}} = \vec{X}_{t,c}] \right), \tag{8}$$

where $\{\omega_{t^{-},c}\}_{t^{-} \in \{1,\ldots,t\}}$ *are fixed weights.*

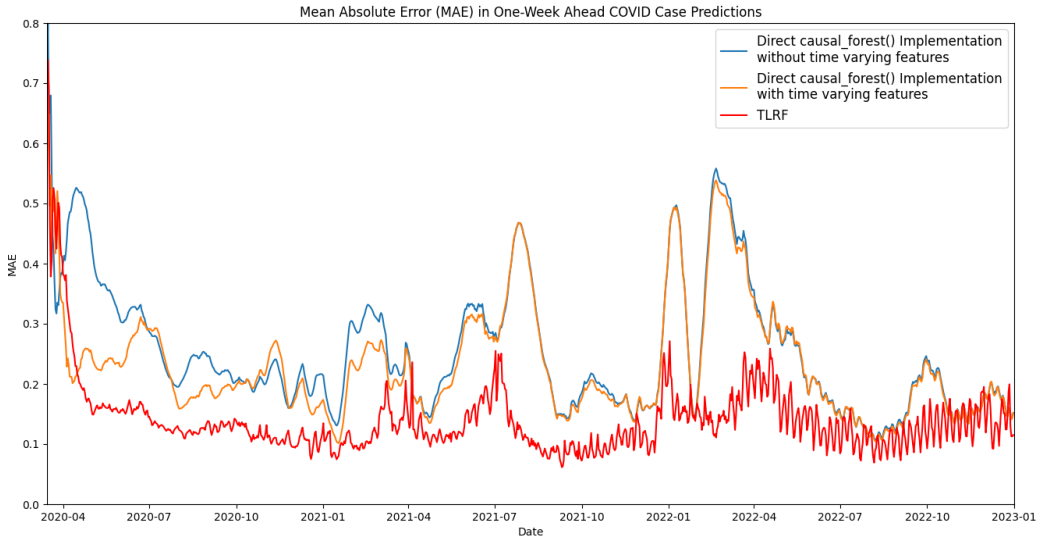

Figure 5 MAE plot of prediction accuracy (TLRF vs. Direct `causal_forest()` Implementation with/without Time-varying Features)

When the feature realization is time-varying, i.e., $\vec{X}_{t',c} \neq \vec{X}_{t,c}$, the conditional expectation $\mathbb{E}[\ln(\boldsymbol{I_{s,c}})|\boldsymbol{s} = t', \vec{\boldsymbol{X}} = \vec{X}_{t,c}]$ in (8) is not identifiable as the relevant data, i.e., $(\ln(I_{t',c}), t', \vec{X}_{t,c})$, is not observable in Table 3. Consequently, incorporating time-varying features essentially compromises the assumption of population overlap, i.e., $\forall t' \in \{1, \ldots, t\}$,

$$0 < \mathbb{P}[\boldsymbol{s} = t', \vec{\boldsymbol{X}} = \vec{X}_{t,c}] < 1, \tag{9}$$

which is a key identification assumption of causal inference methods like `causal_forest()` (D'Amour et al. 2021). Therefore, certain constraints on the incorporated features are necessary to enable a direct `causal_forest()` implementation for county-level instantaneous exponential growth rate estimations.

Notably, our experiment results from Figure 5 and Table 4 suggest that the direct `causal_forest()` implementation inherently constrains features to be time-invariant, i.e., $\forall c \in C$, $\forall t' \in \{1, \ldots, t\}$, $\vec{X}_{t',c} = \vec{X}_c$. As such, the estimator corresponding to the direct `causal_forest()` implementation is obtained by weighing the sample counterpart of CAPE (8) with the GRF similarity measure $\{\alpha'_c(\vec{X}_c)\}_{c' \in C}$, i.e.,

$$\hat{\boldsymbol{r}}^{\boldsymbol{GRF}}_{\boldsymbol{t,c}} = \sum_{c' \in C} \boldsymbol{\alpha_{c'}}(\vec{X}_c) \sum_{t'=2}^{t} \hat{\mu}_{t',c} \left(\ln(\boldsymbol{I_{t',c'}}) - \ln(\boldsymbol{I_{t'-1,c'}})\right), \tag{10}$$

where

$$\boldsymbol{\alpha_{c'}}(\vec{X}_c) := \frac{1}{|B|} \sum_{b=1}^{|B|} \frac{\mathbb{1}(\{\vec{X}_{c'} \in L_b(\vec{X}_c)\})}{|L_b(\vec{X}_c)|}.$$

The formulation (10) reveals that the direct `causal_forest()` implementation is limited in its adaptability. Particularly, the OLS weights $\{\hat{\mu}_{t',c}\}_{t' \in \{2,\ldots,t\}}$ inherited from the CAPE estimate limits the flexibility of the

GRF framework.

Proof of Proposition 1:

By denoting $\forall t^- \in \{t-\delta+2,\ldots,t\}$,

$$\boldsymbol{r}_{t^-,\boldsymbol{c}} := \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t^-,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] - \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t^- -1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}],$$

the CAPE (7) can be rewritten as

$$\begin{aligned}
\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}}),\boldsymbol{s}|\vec{X}_{t,c}] &= \mathbb{E}[(\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}}) - \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])(\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})(\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \mathbb{E}[\mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s},\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}](\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \mathbb{E}[(\boldsymbol{r}_{\boldsymbol{i},\boldsymbol{c}} + \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=i-1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])(\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \mathbb{E}[(\sum_{t^-=t-\delta+2}^{i} \boldsymbol{r}_{t^-,\boldsymbol{c}} + \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t-\delta+1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])(\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}])|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \sum_{i=t-\delta+2}^{t} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\left(\sum_{t^-=t-\delta+2}^{i} \boldsymbol{r}_{t^-,\boldsymbol{c}} + \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t-\delta+1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\right)(i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]) \\
&+ \mathbb{P}[\boldsymbol{s}=t-\delta+1|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t-\delta+1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}](t-\delta+1-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]) \\
&= \sum_{i=t-\delta+2}^{t}\sum_{t^-=t-\delta+2}^{i} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\boldsymbol{r}_{t^-,\boldsymbol{c}}(i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]) \\
&+ \sum_{i=t-\delta+1}^{t} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t-\delta+1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}](i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]) \\
&= \sum_{t^-=t-\delta+2}^{t}\sum_{i=t^-}^{t} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\boldsymbol{r}_{t^-,\boldsymbol{c}}(i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]) \\
&+ \mathbb{E}[\ln(\boldsymbol{I}_{\boldsymbol{s},\boldsymbol{c}})|\boldsymbol{s}=t-\delta+1,\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]\mathbb{E}[\boldsymbol{s}-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}] \\
&= \sum_{t^-=t-\delta+2}^{t} \boldsymbol{r}_{t^-,\boldsymbol{c}} \sum_{i=t^-}^{t} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}](i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]).
\end{aligned}$$

Lastly, by defining $\forall t^- \in \{t-\delta+2,\ldots,t\}$,

$$\omega_{t^-,c} = \sum_{i=t^-}^{t} \mathbb{P}[\boldsymbol{s}=i|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}](i-\mathbb{E}[\boldsymbol{s}|\vec{\boldsymbol{X}}_{\boldsymbol{t},\boldsymbol{c}}]),$$

we have (8). □

Appendix C: Proofs and Additional Theoretical Results

This section provides additional proofs, assumptions, and theoretical justifications for the theorems in the main text. It begins with Appendix C.1, which presents three counterexamples illustrating the role of

Assumptions 1 and 2 in Theorem 1 for the identification purpose. Next, Appendix C.2 includes proofs for the main lemmas and theorems. Following this, Appendix C.3 demonstrates how our identification assumptions are more flexible than the standard OLS assumptions. Finally, Appendix C.4 analyzes how the bias-variance tradeoff for the growth rate estimator depends on the fitting window size.

C.1. Counterexamples

Counterexample 1 *Consider the following specifications of a potential growth model (1):*

$$\ln(\boldsymbol{I}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}}) = 0.5 + 0.75s + \boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}} \tag{11}$$

and

$$\ln(\boldsymbol{I}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}}) = 1 + 0.25s + \boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}}, \tag{12}$$

where $\boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}} := 0.75s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})s$ *and* $\boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}} := 0.25s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})s$*. More precisely, the parameters in this counterexample are specified as* $r(\vec{X}_{2,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{2,c}] = 0.75$*,* $\alpha(\vec{X}_{2,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{2,c}] = 0.5$*,* $r(\vec{X}_{1,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{1,c}] = 0.25$*,* $\alpha(\vec{X}_{1,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{1,c}] = 1$*. It is clear that error terms* $\boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}}$ *and* $\boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}}$ *explicitly depend on* s *and* $\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}}$*.*

We now verify that this specification satisfies ***Assumptions 1****, i.e.,*

$$\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{X}_{t,c}] = \alpha(\vec{X}_{t,c}) + r(\vec{X}_{t,c})s + \mathbb{E}[\boldsymbol{\varepsilon}^{\boldsymbol{t}}_{\boldsymbol{s},\boldsymbol{c}}|\vec{X}_{t,c}] \quad \forall s \in \mathbb{Z}^+,$$

and ***Assumption 2****, i.e.,*

$$\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{X}_{t,c}] = \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t-1}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{X}_{t-1,c}].$$

First, we note that $\forall t^- \in \{1, 2\}$*,*

$$\mathbb{E}[\boldsymbol{\varepsilon}^{\boldsymbol{t^-}}_{\boldsymbol{s},\boldsymbol{c}}|\vec{X}_{t^-,c}] = 0,$$

because

$$\mathbb{E}[\boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}}|\vec{X}_{2,c}] = \mathbb{E}[0.75s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})s|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{2,c}] = 0.75s - 0.75s = 0$$

and

$$\mathbb{E}[\boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}}|\vec{X}_{1,c}] = \mathbb{E}[0.25s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}})s|\vec{\boldsymbol{X}}_{\boldsymbol{s},\boldsymbol{c}} = \vec{X}_{1,c}] = 0.25s - 0.25s = 0$$

As such, (11) and (12) satisfy ***Assumptions 1*** *by their specifications. In addition, they satisfy* ***Assumption 2*** *because*

$$\mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{2}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{X}_{2,c}] = 0.5 + 0.75 \times 1 = 1 + 0.25 \times 1 = \mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{1}}_{\boldsymbol{s},\boldsymbol{c}})|\vec{X}_{1,c}].$$

Counterexample 2 *Consider the following specifications of a potential growth model* (1)*:*

$$\ln(\boldsymbol{I}^{\mathbf{2}}_{\boldsymbol{s,c}}) = 0.5 + 0.75s + \boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s,c}} \tag{13}$$

and

$$\ln(\boldsymbol{I}^{\mathbf{1}}_{\boldsymbol{s,c}}) = 1 + 0.25s + \boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s,c}}, \tag{14}$$

where $\boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s,c}} := 0.75s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})s^2$ *and* $\boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s,c}} := 0.25s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})s$*. More precisely, the parameters in this counterexample are specified as* $r(\vec{X}_{2,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}] = 0.75$*,* $\alpha(\vec{X}_{2,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}] = 0.5$*,* $r(\vec{X}_{1,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{1,c}] = 0.25$*,* $\alpha(\vec{X}_{1,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{1,c}] = 1$*. Clearly, this example satisfies* ***Assumption 2****:*

$$\mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{2}}_{\mathbf{1,c}})|\vec{X}_{2,c}] = 0.5 + 0.75 + (0.75 - 0.75) = 1 + 0.25 + (0.25 - 0.25) = \mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{1}}_{\mathbf{1,c}})|\vec{X}_{1,c}].$$

However, this example does not satisfy ***Assumption 1*** *because the expectation of the potential growth model (13)'s error term conditional on* $\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}$ *is a function of* s*:*

$$\mathbb{E}[\boldsymbol{\varepsilon}^{\mathbf{2}}_{\mathbf{2,c}}|\vec{X}_{2,c}] = \mathbb{E}[0.75s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})s^2|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}] = 0.75s - 0.75s^2.$$

Finally, it is easy to check that ***Theorem 1*** *is not applicable in this example as*

$$r(\vec{X}_{2,c}) = 0.75$$

while

$$\begin{aligned}
&\mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{2}}_{\mathbf{2,c}})|\vec{X}_{2,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\mathbf{1}}_{\mathbf{1,c}})|\vec{X}_{1,c}] \\
&= (0.5 + 0.75 \times 2 + 0.75 \times 2 - 0.75 \times 4) - (1 + 0.25 + (0.25 - 0.25)) \\
&= -0.75.
\end{aligned}$$

Counterexample 3 *Consider the following specifications of a potential growth model* (1)*:*

$$\ln(\boldsymbol{I}^{\mathbf{2}}_{\boldsymbol{s,c}}) = 0.5 + 0.75s + \boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s,c}} \tag{15}$$

and

$$\ln(\boldsymbol{I}^{\mathbf{1}}_{\boldsymbol{s,c}}) = 1 + 0.25s + \boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s,c}}, \tag{16}$$

where $\boldsymbol{\varepsilon}^{\mathbf{2}}_{\boldsymbol{s,c}} := 0.75s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})s$ *and* $\boldsymbol{\varepsilon}^{\mathbf{1}}_{\boldsymbol{s,c}} := 0.25s - r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})s$*. More precisely, the parameters in this counterexample are specified as* $r(\vec{X}_{2,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}] = 0.75$*,* $\alpha(\vec{X}_{2,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{2,c}] = 0.5$*,* $r(\vec{X}_{1,c}) = \mathbb{E}[r(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{1,c}] = 0.25$*,* $\alpha(\vec{X}_{1,c}) = \mathbb{E}[\alpha(\vec{\boldsymbol{X}}_{\boldsymbol{s,c}})|\vec{\boldsymbol{X}}_{\boldsymbol{s,c}} = \vec{X}_{1,c}] = 1$*.*

First, to see that OLS Assumption 2 (OLS Assumption 3 in the original manuscript) is not satisfied by this specification, we note that

$$\mathbb{E}[\boldsymbol{\varepsilon_{2,c}}|\vec{X}_{2,c}] = \mathbb{E}[\boldsymbol{\varepsilon^2_{2,c}}|\vec{X}_{2,c}] = 0 \neq 0.25 - 0.75 = \mathbb{E}[\boldsymbol{\varepsilon^1_{1,c}}|\vec{X}_{2,c}] = \mathbb{E}[\boldsymbol{\varepsilon_{1,c}}|\vec{X}_{2,c}].$$

Subsequently, we verify that this specification satisfies ***Assumptions 1****, i.e.,*

$$\mathbb{E}[\ln(\boldsymbol{I^t_{s,c}})|\vec{X}_{t,c}] = \alpha(\vec{X}_{t,c}) + r(\vec{X}_{t,c})s + \mathbb{E}[\boldsymbol{\varepsilon^t_{s,c}}|\vec{X}_{t,c}] \quad \forall s \in \mathbb{Z}^+,$$

and ***Assumption 2****, i.e.,*

$$\mathbb{E}[\ln(\boldsymbol{I^t_{s,c}})|\vec{X}_{t,c}] = \mathbb{E}[\ln(\boldsymbol{I^{t-1}_{s,c}})|\vec{X}_{t-1,c}].$$

First, we note that $\forall t^- \in \{1, 2\}$,

$$\mathbb{E}[\boldsymbol{\varepsilon^{t^-}_{s,c}}|\vec{X}_{t^-,c}] = 0,$$

because

$$\mathbb{E}[\boldsymbol{\varepsilon^2_{s,c}}|\vec{X}_{2,c}] = \mathbb{E}[0.75s - r(\boldsymbol{\vec{X}_{s,c}})s|\boldsymbol{\vec{X}_{s,c}} = \vec{X}_{2,c}] = 0.75s - 0.75s = 0$$

and

$$\mathbb{E}[\boldsymbol{\varepsilon^1_{s,c}}|\vec{X}_{1,c}] = \mathbb{E}[0.25s - r(\boldsymbol{\vec{X}_{s,c}})s|\boldsymbol{\vec{X}_{s,c}} = \vec{X}_{1,c}] = 0.25s - 0.25s = 0.$$

Lastly, this specification satisfies ***Assumption 2*** *because*

$$\mathbb{E}[\ln(\boldsymbol{I^2_{s,c}})|\vec{X}_{2,c}] = 0.5 + 0.75 \times 1 = 1 + 0.25 \times 1 = \mathbb{E}[\ln(\boldsymbol{I^1_{s,c}})|\vec{X}_{1,c}].$$

C.2. Proofs for Results Stated in the Main Text

Proof of Lemma 1:

$$\begin{aligned} r_{t,c} &= r(\vec{X}_{t,c}) \\ &= \left(\alpha(\vec{X}_{t,c}) + r(\vec{X}_{t,c})t + \mathbb{E}[\boldsymbol{\varepsilon^t_{t,c}}|\vec{X}_{t,c}]\right) - \left(\alpha(\vec{X}_{t,c}) + r(\vec{X}_{t,c})(t-1) + \mathbb{E}[\boldsymbol{\varepsilon^t_{t,c}}|\vec{X}_{t,c}]\right) \\ &= \mathbb{E}\left[\ln(\boldsymbol{I^t_{t,c}}) - \ln(\boldsymbol{I^t_{t-1,c}})\middle|\vec{X}_{t,c}\right] \quad \text{(by Assumption 1).} \end{aligned}$$

□

Proof of Theorem 1:

We aim to establish the equality (4) between the county-level instantaneous exponential growth rate and the observed log case number changes, i.e.,

$$r(\vec{X}_{t,c}) = \mathbb{E}[\ln(\boldsymbol{I^t_{t,c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I^{t-1}_{t-1,c}})|\vec{X}_{t-1,c}].$$

To this end, we utilize the following decomposition:

$$\begin{aligned}
&\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t},\boldsymbol{c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t-1}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t-1,c}] \\
=&(\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t},\boldsymbol{c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t,c}]) \\
+&(\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t-1}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t-1,c}]).
\end{aligned} \tag{17}$$

By Lemma 1, the county-level instantaneous exponential growth rate $r_{t,c}$ is given by the first term in the decomposition (17), i.e.,

$$r_{t,c} = \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t},\boldsymbol{c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t,c}].$$

The second term in the decomposition (17) is 0, i.e.,

$$0 = \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t-1}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t-1,c}],$$

by Assumption 2 (Overlap). Therefore, the decomposition (17) establishes the desired equality (4) under Assumptions 1 and 2. □

Proof of Corollary 1:

The conditional mean outcome (8) is obtained by plugging in

$$\mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t,c}] = \mathbb{E}[\ln(\boldsymbol{I}^{\boldsymbol{t-1}}_{\boldsymbol{t-1},\boldsymbol{c}})|\vec{X}_{t-1,c}]$$

from Assumption 2 (Overlap) into the equality (4) established by Theorem 1. The initial estimator is the sample counterpart of the conditional mean outcome (8). □

Proof of Theorem 2:

Following §6.1 by Athey et al. (2019), the estimator derived from the weighted conditional mean outcome (11), i.e.,

$$r(\vec{X}_{t,c}) = \sum_{t' \in [t]} \sum_{c' \in C} \gamma_{t',c'}(X_{t,c}) \mathbb{E}[(\ln(\boldsymbol{I}_{\boldsymbol{t',c'}}) - \ln(\boldsymbol{I}_{\boldsymbol{t'-1,c'}}))|\vec{X}_{t,c}],$$

is a consistent estimator of $r_{t,c}$ provided that $\forall t' \in [t]$, $\forall c' \in C$,

$$\mathbb{E}[(\ln(\boldsymbol{I}_{\boldsymbol{t',c'}}) - \ln(\boldsymbol{I}_{\boldsymbol{t'-1,c'}}))|\vec{X}_{t,c}]$$

is Lipschitz continuous in $\vec{X}_{t,c}$. Clearly, this condition follows from Assumptions 2 and 3. In addition, the sample counterpart of (11) is (12), i.e.,

$$\hat{\boldsymbol{r}}^{\boldsymbol{TLRF}}_{\boldsymbol{t,c}} = \sum_{t' \in [t]} \sum_{c' \in C} \gamma_{t',c'}(\vec{\boldsymbol{X}}_{\boldsymbol{t,c}}) \left(\ln(\boldsymbol{I}_{\boldsymbol{t',c'}}) - \ln(\boldsymbol{I}_{\boldsymbol{t'-1,c'}})\right).$$

□

Proof of Proposition 1:

$$
\begin{aligned}
&\hat{\boldsymbol{r}}^{\boldsymbol{RF}}_{\boldsymbol{t,c}} \\
=&\frac{1}{|B|}\sum_{b=1}^{|B|}\sum_{t'\in[t]}\sum_{c'\in C}(\ln(\boldsymbol{I_{t',c'}})-\ln(\boldsymbol{I_{t'-1,c'}}))\frac{\mathbb{1}(\{\vec{X}_{t',c'}\in L_b(\vec{X}_{t,c})\})}{|L_b(\vec{X}_{t,c})|} \\
=&\sum_{t'\in[t]}\sum_{c'\in C}\frac{1}{|B|}\sum_{b=1}^{|B|}\frac{\mathbb{1}(\{\vec{\boldsymbol{X}}_{\boldsymbol{t',c'}}\in L_b(\vec{X}_{t,c})\})}{|L_b(\vec{X}_{t,c})|}\left(\ln(\boldsymbol{I_{t',c'}})-\ln(\boldsymbol{I_{t'-1,c'}})\right) \\
=&\sum_{t'\in[t]}\sum_{c'\in C}\boldsymbol{\gamma_{t',c'}}(\vec{X}_{t,c})\left(\ln(\boldsymbol{I_{t',c'}})-\ln(\boldsymbol{I_{t'-1,c'}})\right) \\
=&\hat{\boldsymbol{r}}^{\boldsymbol{TLRF}}_{\boldsymbol{t,c}}.
\end{aligned}
$$

□

C.3. OLS Identification Assumptions of $r_{t,c}$

This section derives the OLS estimators of $\boldsymbol{r_{t,c}}$ for fixed fitting window sizes under standard OLS identification assumptions, as outlined in Theorem 2. Following this, Theorem 3 highlights that the identification assumptions we employ facilitate the development of more versatile estimators of $\boldsymbol{r_{t,c}}$ in comparison to those derived from the OLS assumptions. Within this section, we employ the abbreviations $\boldsymbol{\alpha}$ and $\boldsymbol{r}$ to represent the random coefficients $\alpha(\vec{\boldsymbol{X}})$ and $r(\vec{\boldsymbol{X}})$, correspondingly, i.e.,

$$\boldsymbol{\alpha} := \alpha(\vec{\boldsymbol{X}}) \text{ and } \boldsymbol{r} := r(\vec{\boldsymbol{X}}).$$

The conditional expectations of these coefficients, given the relevant features at county c on day t, are denoted as $\alpha(\vec{X}_{t,c})$ and $r(\vec{X}_{t,c})$, respectively, i.e.,

$$\alpha(\vec{X}_{t,c}) := \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] \text{ and } r(\vec{X}_{t,c}) := \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}].$$

We first state the identification assumptions for OLS estimators:

OLS Assumption 1 *(Homogeneity)* $\forall t^- \in \{t-\delta+1,\ldots,t\}$,

$$\alpha(\vec{X}_{t,c}) = \alpha(\vec{X}_{t^-,c}),\ \ r(\vec{X}_{t,c}) = r(\vec{X}_{t^-,c}),\ \ \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] = \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t^-,c}];$$

OLS Assumption 2 *(Normalization)* $\forall t^- \in \{t-\delta+1,\ldots,t\}$,

$$\mathbb{E}[\boldsymbol{\varepsilon_{t,c}}|\vec{X}_{t,c}] = \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}].$$

Subsequently, we demonstrate Theorem 2 that the identification of $r_{t,c}$ is also possible through these assumptions:

THEOREM 2. *Let $\delta \in \mathbb{Z}^+ \backslash \{1\}$ be a fixed fitting window size, and $c \in C$ be a fixed county. The county-level instantaneous exponential growth rate $r_{t,c}$ is identified as*

$$r_{t,c} = \frac{\sum_{t^-=t-\delta+1}^{t} (t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta})}{\sum_{t^-=t-\delta+1}^{t} \left(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta}\right)^2}. \tag{18}$$

Proof of Theorem 2:

Our goal is to identify $r_{t,c}$ using the log case data $\{ln(I_{t^-,c})\}_{t^- \in \{t-\delta+1,\ldots,t\}}$ from county c within the fitting window of size δ. To achieve this, we first observe that by OLS Assumption 1, the actual growth model (2) reduces to a homogeneous growth model, i.e., $\forall t^- \in \{t-\delta+1,\ldots,t\}$,

$$\ln(\boldsymbol{I_{t^-,c}}) = \alpha_{t,c} + r_{t,c} t^- + \boldsymbol{\varepsilon_{t^-,c}}.$$

By OLS Assumptions 1 and 2, we have

$$\begin{aligned} &\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}}) - \boldsymbol{\alpha} - \boldsymbol{r} t^- - \boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] = 0 \quad \forall t^- \in \{t-\delta+1,\ldots,t\} \\ \Rightarrow &\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] = \sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] + \sum_{t^-=t-\delta+1}^{t} (t^-)^2 \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] + \sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}]. \end{aligned} \tag{19}$$

By subtracting

$$\begin{aligned} \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- = \frac{1}{\delta} \Bigg(&\delta \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- \\ &+ \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] \left(\sum_{t^-=t-\delta+1}^{t} t^- \right)^2 + \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- \Bigg) \end{aligned}$$

from (19), we have

$$\begin{aligned} &\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- \\ = &\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] - \frac{1}{\delta} \delta \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- \\ + &\sum_{t^-=t-\delta+1}^{t} (t^-)^2 \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] - \delta \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] \left(\sum_{t^-=t-\delta+1}^{t} \frac{t^-}{\delta} \right)^2 \\ + &\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] - \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^-. \end{aligned} \tag{20}$$

Finally, (20) implies (18) because

$$\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^-$$

$$= \sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \delta \sum_{t^-=t-\delta+1}^{t} \frac{\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta} \sum_{t^-=t-\delta+1}^{t} \frac{t^-}{\delta}$$

$$= \sum_{t^-=t-\delta+1}^{t} (t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta}), \tag{21}$$

$$\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] - \frac{1}{\delta}\delta\mathbb{E}[\boldsymbol{\alpha}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- = 0,$$

$$\sum_{t^-=t-\delta+1}^{t} (t^-)^2 \mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] - \delta\mathbb{E}[\boldsymbol{r}|\vec{X}_{t,c}] \left(\sum_{t^-=t-\delta+1}^{t} \frac{t^-}{\delta} \right)^2 = \sum_{t^-=t-\delta+1}^{t} \left(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta} \right)^2 r_{t,c}, \tag{22}$$

and

$$\sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] - \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\boldsymbol{\varepsilon_{t^-,c}}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^-$$

$$= \sum_{t^-=t-\delta+1}^{t} t^- \mathbb{E}[\boldsymbol{\varepsilon_{t,c}}|\vec{X}_{t,c}] - \frac{1}{\delta} \sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\boldsymbol{\varepsilon_{t,c}}|\vec{X}_{t,c}] \sum_{t^-=t-\delta+1}^{t} t^- \qquad \text{(by OLS Assumption 2)}$$

$$=0,$$

where (21) and (22) make use of the fact that, for $\{(x_i, y_i) \in \mathbb{R}^2 | i = 1, 2, \ldots, n\}$, we have

$$\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}.$$

□

Subsequently, we show that our identification assumptions 1 and 2 allow for more adaptive estimators of the instantaneous county-level exponential growth rate compared to those derived from the OLS Assumptions 1 and 2.

THEOREM 3. *Any OLS estimators (18) of $r_{t,c}$ with a δ-day fitting window can be expressed as a convex combination of $\delta - 1$ two-point estimators in the form (4), i.e., $\forall \delta \in \mathbb{Z}^+ \backslash \{1\}$,*

$$\frac{\sum_{t^-=t-\delta+1}^{t} (t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t} \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta})}{\sum_{t^-=t-\delta+1}^{t} \left(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta} \right)^2}$$

$$= \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I_{t^--1,c}})|\vec{X}_{t,c}] \right), \tag{23}$$

where $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ are non-negative weights that sum to 1, i.e.,

$$\hat{\mu}_{t^-,c} = \frac{\sum_{i=t^-}^{t} (i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t} (i - (t-\delta+1))(i - \frac{2t-\delta+1}{2})}. \tag{24}$$

As such, the sample counterpart of the OLS estimator (18) with a δ-day fitting window is

$$\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}^{\boldsymbol{ols}(\delta)} = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I_{t^-,c}}) - \ln(\boldsymbol{I_{t^--1,c}}) \right). \tag{25}$$

Proof of Theorem 3:

First, the numerator of the OLS estimator (18) can be rewritten as

$$\begin{aligned}
&\sum_{t^-=t-\delta+1}^{t}(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t}\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta}) \\
&= \sum_{t^-=t-\delta+1}^{t}\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}](t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})
\end{aligned} \tag{26}$$

Additionally, by rewriting $r_{t^-,c}$ as (4), i.e., $\forall t^- \in \{t-\delta+2,\dots,t\}$,

$$r_{t^-,c} = \mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \mathbb{E}[\ln(\boldsymbol{I_{t^--1,c}})|\vec{X}_{t,c}], \tag{27}$$

we have $\forall \delta \in \mathbb{Z}^+\backslash\{1\}$, $\forall i \in \{t-\delta+2,\dots,t\}$,

$$\begin{aligned}
\mathbb{E}[\ln(\boldsymbol{I_{i,c}})|\vec{X}_{t,c}] &= r_{i,c} + \mathbb{E}[\ln(\boldsymbol{I_{i-1,c}})|\vec{X}_{t,c}] \\
&= \sum_{t^-=t-\delta+2}^{i} r_{t^-,c} + \mathbb{E}[\ln(\boldsymbol{I_{t-\delta+1,c}})|\vec{X}_{t,c}].
\end{aligned} \tag{28}$$

By (28), (26) becomes

$$\begin{aligned}
&\sum_{t^-=t-\delta+1}^{t}(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t}\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta}) \\
&= \sum_{i=t-\delta+2}^{t}\left(\sum_{t^-=t-\delta+2}^{i} r_{t^-,c} + \mathbb{E}[\ln(\boldsymbol{I_{t-\delta+1,c}})|\vec{X}_{t,c}]\right)(i - \frac{\sum_{i=t-\delta+1}^{t} i}{\delta}) \\
&+ \mathbb{E}[\ln(\boldsymbol{I_{t-\delta+1,c}})|\vec{X}_{t,c}](t-\delta+1 - \frac{\sum_{i=t-\delta+1}^{t} i}{\delta}) \\
&= \sum_{i=t-\delta+2}^{t}\sum_{t^-=t-\delta+2}^{i} r_{t^-,c}(i - \frac{2t-\delta+1}{2}) + \sum_{i=t-\delta+1}^{t}\mathbb{E}[\ln(\boldsymbol{I_{t-\delta+1,c}})|\vec{X}_{t,c}](i - \frac{2t-\delta+1}{2}) \\
&= \sum_{t^-=t-\delta+2}^{t}\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2}) r_{t^-,c}
\end{aligned}$$

Therefore, the OLS estimator (18) can be written as

$$\begin{aligned}
&\frac{\sum_{t^-=t-\delta+1}^{t}(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta})(\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}] - \frac{\sum_{t^-=t-\delta+1}^{t}\mathbb{E}[\ln(\boldsymbol{I_{t^-,c}})|\vec{X}_{t,c}]}{\delta})}{\sum_{t^-=t-\delta+1}^{t}\left(t^- - \frac{\sum_{t^-=t-\delta+1}^{t} t^-}{\delta}\right)^2} \\
&= \sum_{t^-=t-\delta+2}^{t}\frac{\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}\left(i - \frac{\sum_{i=t-\delta+1}^{t} i}{\delta}\right)^2} r_{t^-,c} \\
&= \sum_{t^-=t-\delta+2}^{t}\frac{\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t} i\left(i - \frac{2t-\delta+1}{2}\right)} r_{t^-,c} \\
&= \sum_{t^-=t-\delta+2}^{t}\frac{\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+2}^{t}(i-(t-\delta+1))\left(i - \frac{2t-\delta+1}{2}\right)} r_{t^-,c}.
\end{aligned}$$

As such, by defining $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ and $\{r_{t^-,c,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ according to (24) and (27), we have (23) and its sample counterpart (25). Notably, the sample weights (24) are fixed and independent of feature realizations.

Lastly, we show that $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ are non-negative weights that sum to 1. First, $\sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} = 1$ because

$$\begin{aligned}
\sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} &= \sum_{t^-=t-\delta+2}^{t} \frac{\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+2}^{t}(i-(t-\delta+1))\left(i-\frac{2t-\delta+1}{2}\right)} \\
&= \frac{\sum_{i=t-\delta+2}^{t}\sum_{t^-=t-\delta+2}^{i}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+2}^{t}(i-(t-\delta+1))\left(i-\frac{2t-\delta+1}{2}\right)} \\
&= \frac{\sum_{i=t-\delta+2}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+2}^{t}(i-(t-\delta+1))\left(i-\frac{2t-\delta+1}{2}\right)} \\
&= 1.
\end{aligned}$$

Second, to see that $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ are non-negative, we first demonstrate that their numerators are non-negative for any realized features $x_{t,c}$.

Case 1: $\forall t^- \in \{t-\delta+2,\ldots,t\}$ where $t^- - 1 \leq \frac{2t-\delta+1}{2}$, the numerator of $\hat{\mu}_{t^-,c}$ can be expressed as

$$\begin{aligned}
&\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2}) \\
=&\sum_{i=t-\delta+1}^{t}\left(i-\frac{2t-\delta+1}{2}\right) - \sum_{i=t-\delta+1}^{t^--1}\left(i-\frac{2t-\delta+1}{2}\right) \\
=&\sum_{i=t-\delta+1}^{t^--1}\left(\frac{2t-\delta+1}{2}-i\right) \\
\geq& 0,
\end{aligned}$$

which is non-negative.

Case 2: $\forall t^- \in \{t-\delta+2,\ldots,t\}$ where $t^- - 1 > \frac{2t-\delta+1}{2}$, the numerator of $\hat{\mu}_{t^-,c}$ is positive, i.e.,

$$\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2}) > 0.$$

By combining Case 1 and Case 2, we thus prove that the numerator of each $\hat{\mu}_{t^-,c}$ is non-negative for any $t^- \in \{t-\delta+2,\ldots,t\}$.

In addition, since $\sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} = 1$, the denominator of each $\hat{\mu}_{t^-,c}$, as a sum of non-negative numerators, is also non-negative. Hence $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$ are non-negative. □

Lastly, we state and prove a useful lemma that clarifies the distribution of OLS weights (24).

LEMMA 2. *The OLS weights* $\{\hat{\mu}_{t^-,c}\}_{t^-\in\{t-\delta+2,\dots,t\}}$, *where*

$$\hat{\mu}_{t^-,c} = \frac{\sum_{i=t^-}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})},$$

are concentrated towards the center, i.e.,

$\hat{\mu}_{t,c} = 1$ *when* $\delta = 2$, $\hat{\mu}_{t,c} = \hat{\mu}_{t-1,c}$ *when* $\delta = 3$, *and*

$$\hat{\mu}_{t,c} < \cdots < \hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor,c} = \hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil,c} > \cdots > \hat{\mu}_{t-\delta+2,c}$$

when $\delta > 3$.

Proof of Lemma 2:

The first two claims, i.e., $\hat{\mu}_{t,c} = 1$ when $\delta = 2$, $\hat{\mu}_{t,c} = \hat{\mu}_{t-1,c}$ when $\delta = 3$, can be directly verified.

To see that the third claim is true, we first verify $\hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor,c} = \hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil,c}$. When δ is an even number, we have $\lfloor\frac{2t-\delta+2}{2}\rfloor = \lceil\frac{2t-\delta+2}{2}\rceil$, which directly implies $\hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor,c} = \hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil,c}$. When δ is an odd number, we have

$$\begin{aligned}
\hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil,c} =& \frac{\sum_{i=\lceil\frac{2t-\delta+2}{2}\rceil}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
=& \frac{\sum_{i=\lfloor\frac{2t-\delta+2}{2}\rfloor}^{t}(i-\frac{2t-\delta+1}{2}) - (\frac{2t-\delta+1}{2}-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
=& \frac{\sum_{i=\lfloor\frac{2t-\delta+2}{2}\rfloor}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
=& \hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor,c}.
\end{aligned}$$

Additionally, for all $\eta \in \mathbb{Z}^+$ such that $t-\delta+2 < \lfloor\frac{2t-\delta+2}{2}\rfloor - \eta < \lceil\frac{2t-\delta+2}{2}\rceil + \eta < t$, we have $\hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil+\eta,c} > \hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil+\eta+1,c}$, i.e.,

$$\begin{aligned}
\hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil+\eta+1,c} =& \frac{\sum_{i=\lceil\frac{2t-\delta+2}{2}\rceil+\eta+1}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
=& \frac{\sum_{i=\lceil\frac{2t-\delta+2}{2}\rceil+\eta}^{t}(i-\frac{2t-\delta+1}{2}) - (\lceil\frac{2t-\delta+2}{2}\rceil+\eta+1-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
<& \frac{\sum_{i=\lceil\frac{2t-\delta+2}{2}\rceil+\eta}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})} \\
=& \hat{\mu}_{\lceil\frac{2t-\delta+2}{2}\rceil+\eta,c},
\end{aligned}$$

and $\hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor-\eta,c} > \hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor-\eta-1,c}$, i.e.,

$$\hat{\mu}_{\lfloor\frac{2t-\delta+2}{2}\rfloor-\eta-1,c} = \frac{\sum_{i=\lfloor\frac{2t-\delta+2}{2}\rfloor-\eta-1}^{t}(i-\frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})}$$

$$
\begin{aligned}
&= \frac{\sum_{i=\lfloor \frac{2t-\delta+2}{2} \rfloor - \eta}^{t} (i - \frac{2t-\delta+1}{2}) + (\lfloor \frac{2t-\delta+2}{2} \rfloor - \eta - 1 - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t} (i-(t-\delta+1))(i - \frac{2t-\delta+1}{2})} \\
&< \frac{\sum_{i=\lfloor \frac{2t-\delta+2}{2} \rfloor - \eta}^{t} (i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t} (i-(t-\delta+1))(i - \frac{2t-\delta+1}{2})} \\
&= \hat{\mu}_{\lfloor \frac{2t-\delta+2}{2} \rfloor - \eta, c}.
\end{aligned}
$$

As such, we have shown that for $\delta > 3$,

$$
\hat{\mu}_{t,c} < \cdots < \hat{\mu}_{\lfloor \frac{2t-\delta+2}{2} \rfloor, c} = \hat{\mu}_{\lceil \frac{2t-\delta+2}{2} \rceil, c} > \cdots > \hat{\mu}_{t-\delta+2,c}.
$$

□

C.4. Analytical Results of the Bias-variance Trade-off for Instantaneous Growth Rate Estimations

This section examines how the choice of fitting window size affects the bias-variance trade-off in estimating instantaneous growth rates.

First, we obtain the standard bias-variance decomposition of an estimator's MSE:

THEOREM 4. *For any estimator $\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}$ of the county-level instantaneous exponential growth rate $r(\vec{X}_{t,c})$, its Mean-squared error (MSE) has the following bias-variance decomposition:*

$$
\mathbb{E}[\left(r(\vec{X}_{t,c}) - \hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}\right)^2] = \underbrace{(r(\vec{X}_{t,c}) - \mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}])^2}_{\textbf{\textit{Bias}}} + \underbrace{\mathbb{V}\text{ar}[\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}]}_{\textbf{\textit{Variance}}}. \tag{29}
$$

Second, We provide an analytical result for how the bias-variance trade-off of OLS estimators is affected by their fitting window sizes:

COROLLARY 1. *For an OLS estimator with δ-day fitting window ($\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{ols}(\delta)}$), its MSE, i.e.,*

$$
\begin{aligned}
&\mathbb{E}[\left(r(\vec{X}_{t,c}) - \hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{ols}(\delta)}\right)^2] \\
&= \left(\underbrace{r(\vec{X}_{t,c}) - \mathbb{E}[\sum_{t^- = t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I}_{\boldsymbol{t^-},\boldsymbol{c}}) - \ln(\boldsymbol{I}_{\boldsymbol{t^- - 1},\boldsymbol{c}})\right) | \vec{X}_{t,c}]}_{\textbf{\textit{Bias}}} \right)^2 \\
&+ \underbrace{\mathbb{V}\text{ar}[\sum_{t^- = t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I}_{\boldsymbol{t^-},\boldsymbol{c}}) - \ln(\boldsymbol{I}_{\boldsymbol{t^- - 1},\boldsymbol{c}})\right) | \vec{X}_{t,c}]}_{\textbf{\textit{Variance}}},
\end{aligned} \tag{30}
$$

has a zero bias term when $\delta = 2$ and has an asymptotically vanishing variance term when $\delta = t$.

Notably, changing the fitting window size δ cannot strike the right balance for the bias-variance tradeoff in (30) due to the non-adaptive weighting scheme $\{\hat{\mu}_{t^-,c}\}_{t^- \in \{t-\delta+2,\ldots,t\}}$.

In contrast, through adaptive weighting, the TLRF estimator $\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}$ can more effectively balance the bias and variance tradeoff in its MSE:

$$
\begin{aligned}
&\mathbb{E}[\left(r(\vec{X}_{t,c}) - \hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}\right)^2] \\
&= \Bigg(\underbrace{r(\vec{X}_{t,c}) - \mathbb{E}[\sum_{t' \in [t]} \sum_{c' \in C} \boldsymbol{\gamma}_{\boldsymbol{t',c'}}(\vec{X}_{t,c}) \left(\ln(\boldsymbol{I}_{\boldsymbol{t',c'}}) - \ln(\boldsymbol{I}_{\boldsymbol{t'-1,c'}})\right) | \vec{X}_{t,c}]}_{\textbf{Bias}} \Bigg)^2 \\
&+ \underbrace{\mathbb{V}\mathbb{a}\mathbb{r}[\sum_{t' \in [t]} \sum_{c' \in C} \boldsymbol{\gamma}_{\boldsymbol{t',c'}}(\vec{X}_{t,c}) \left(\ln(\boldsymbol{I}_{\boldsymbol{t',c'}}) - \ln(\boldsymbol{I}_{\boldsymbol{t'-1,c'}})\right) | \vec{X}_{t,c}]}_{\textbf{Variance}} .
\end{aligned} \tag{31}
$$

That is, by selecting $\{\boldsymbol{\gamma}_{\boldsymbol{t',c'}}(\vec{X}_{t,c})\}_{t' \in [t], c' \in C}$ to minimize (31), one can achieve better or at least equal MSE compared to selecting δ to minimize (30). More precisely, (31) would yield the same value as (30) if we choose the weights $\boldsymbol{\gamma}_{\boldsymbol{t',c'}}(X_{t,c}) = \hat{\mu}_{t',c} \quad \forall t' \in \{t-\delta+2,\ldots,t\}$ and $\boldsymbol{\gamma}_{\boldsymbol{t',c'}}(X_{t,c}) = 0 \quad \forall t' \in \{2,\ldots,t-\delta+1\}$ or $\forall c' \in C \backslash \{c\}$. The key insight here is that TLRF offers a generalized approach to solving the fitting window selection problem by transforming it into a weight assignment problem.

Third, we discuss how TLRF manages its bias-variance trade-off. Notably, balancing the bias-variance trade-off for TLRF essentially involves choosing $\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}$ to minimize (31). However, since the ground truth, i.e., growth rate $r(\vec{X}_{t,c})$, is not observable, optimizing (31) empirically requires certain transformations. For example, choosing $\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}$ to minimize (31) is equivalent to choosing $\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}$ to maximize

$$
r^2(\vec{X}_{t,c}) - \mathbb{E}[\left(r(\vec{X}) - \hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}\right)^2]. \tag{32}
$$

Athey and Imbens (2016) observed that if $\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}}] = r(\vec{X}_{t,c})$, then the sample counterpart of (32) reduces to $(\hat{\boldsymbol{r}}_{\boldsymbol{t},\boldsymbol{c}}^{\boldsymbol{TLRF}})^2$, which is directly measurable and thus can be optimized empirically. Subsequently, Athey et al. (2019) further relaxed this assumption and demonstrated that their GRF method achieves consistency if the parameter of interest is identifiable by a conditional mean outcome. Therefore, to balance the bias-variance trade-off of TLRF, we first find a conditional mean outcome that identifies $r(\vec{X}_{t,c})$ in §4.1 and subsequently apply the random forest algorithm to obtain a consistent final estimator in §4.2.

Proofs for this section:

Proof of Theorem 4:

$$
\begin{aligned}
&\mathbb{E}[\left(r(\vec{X}_{t,c}) - \hat{\boldsymbol{r}}_{\boldsymbol{t,c}}\right)^2] \\
&= r^2(x_{t,c}) - 2r(x_{t,c})\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}] + \mathbb{E}[\hat{\boldsymbol{r}}^{\boldsymbol{2}}_{\boldsymbol{t,c}}] \\
&= r^2(x_{t,c}) - 2r(x_{t,c})\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}] + \mathbb{E}[\hat{\boldsymbol{r}}^{\boldsymbol{2}}_{\boldsymbol{t,c}}] + (\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}])^2 - (\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}])^2 \\
&= r^2(x_{t,c}) - 2r(x_{t,c})\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}] + (\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}])^2 + \mathbb{E}[\hat{\boldsymbol{r}}^{\boldsymbol{2}}_{\boldsymbol{t,c}}] - (\mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}])^2 \\
&= \underbrace{(r(x_{t,c}) - \mathbb{E}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}])^2}_{\textbf{Bias}} + \underbrace{\mathbb{V}\mathbb{a}\mathbb{r}[\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}]}_{\textbf{Variance}}.
\end{aligned}
$$

□

Proof of Corollary 1:

By Theorem 3, an OLS estimator with δ-day fitting window can be expressed as:

$$
\hat{\boldsymbol{r}}^{\boldsymbol{ols}(\delta)}_{\boldsymbol{t,c}} = \sum_{t^-=t-\delta+2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I_{t^-,c}}) - \ln(\boldsymbol{I_{t^--1,c}})\right),
$$

where

$$
\hat{\mu}_{t^-,c} = \frac{\sum_{i=t^-}^{t}(i - \frac{2t-\delta+1}{2})}{\sum_{i=t-\delta+1}^{t}(i-(t-\delta+1))(i-\frac{2t-\delta+1}{2})}.
$$

First, when $\delta = 2$, the bias term of its MSE (29), i.e.,

$$
r(\vec{X}_{t,c}) - \mathbb{E}[\ln(\boldsymbol{I_{t,c}}) - \ln(\boldsymbol{I_{t-1,c}})|\vec{X}_{t,c}],
$$

is zero by Theorem 1.

Second, when $\delta = t$, we show that the variance term of its MSE (29) vanishes asymptotically, i.e.,

$$
lim_{t\to\infty}\mathbb{V}\mathbb{a}\mathbb{r}[\hat{\boldsymbol{r}}^{\boldsymbol{ols}(\delta)}_{\boldsymbol{t,c}}] = 0. \tag{33}
$$

To see this, We first note that the variance term can be upper-bounded in the following fashion:

$$
\begin{aligned}
&\mathbb{V}\mathbb{a}\mathbb{r}[\hat{\boldsymbol{r}}^{\boldsymbol{ols}(\delta)}_{\boldsymbol{t,c}}] \\
=&\mathbb{V}\mathbb{a}\mathbb{r}[\sum_{t^-=2}^{t} \hat{\mu}_{t^-,c} \left(\ln(\boldsymbol{I_{t^-,c}}) - \ln(\boldsymbol{I_{t^--1,c}})\right)|\vec{X}_{t,c}] \\
=&\sum_{t'=2}^{t}\sum_{t''=2}^{t} \hat{\mu}_{t',c}\hat{\mu}_{t'',c}\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I_{t',c}}) - \ln(\boldsymbol{I_{t'-1,c}}), \ln(\boldsymbol{I_{t'',c}}) - \ln(\boldsymbol{I_{t''-1,c}})|\vec{X}_{t,c}] \\
\leq&\sum_{t'=2}^{t}\sum_{t''=2}^{t} (\hat{\mu}_{\lfloor\frac{t+2}{2}\rfloor,c})^2 \left|\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I_{t',c}}) - \ln(\boldsymbol{I_{t'-1,c}}), \ln(\boldsymbol{I_{t'',c}}) - \ln(\boldsymbol{I_{t''-1,c}})|\vec{X}_{t,c}]\right| \qquad (34) \\
=&(\hat{\mu}_{\lfloor\frac{t+2}{2}\rfloor,c})^2 \sum_{t'=2}^{t}\sum_{t''=2}^{t} \left|\mathbb{C}\mathbb{o}\mathbb{v}[\ln(\boldsymbol{I_{t',c}}) - \ln(\boldsymbol{I_{t'-1,c}}), \ln(\boldsymbol{I_{t'',c}}) - \ln(\boldsymbol{I_{t''-1,c}})|\vec{X}_{t,c}]\right|, \qquad (35)
\end{aligned}
$$

where (34) follows from Lemma 2, i.e.,

$$\hat{\mu}_{t,c} < \cdots < \hat{\mu}_{\lfloor \frac{t+2}{2} \rfloor,c} = \hat{\mu}_{\lceil \frac{t+2}{2} \rceil,c} > \cdots > \hat{\mu}_{t-\delta+2,c}.$$

Therefore, to show (33), it suffices to show (35) approaches zero as t approaches infinity.

In addition, as a sufficient condition for ergodicity of second moments, it is common to assume that the autocovariance is absolutely summable, i.e.,

$$lim_{t\to\infty} \sum_{t'=2}^{t} \sum_{t''=2}^{t} \left| \mathbb{C}\text{ov}[\ln(\boldsymbol{I_{t',c}}) - \ln(\boldsymbol{I_{t'-1,c}}), \ln(\boldsymbol{I_{t'',c}}) - \ln(\boldsymbol{I_{t''-1,c}}) | \vec{X}_{t,c}] \right| < \infty,$$

when proving limit theorems for serially dependent observations (Hamilton 2020).

As such, to show (35) approaches zero as t approaches infinity, it suffices to show that

$$lim_{t\to\infty} (\hat{\mu}_{\lfloor \frac{t+2}{2} \rfloor,c})^2 = 0. \tag{36}$$

In fact, by fixing the fitting window size $\delta = t$, we have

$$\begin{aligned}
\hat{\mu}_{\lfloor \frac{t+2}{2} \rfloor,c} &= \frac{\sum_{i=\lfloor \frac{t+2}{2} \rfloor}^{t} (i - \frac{2t-t+1}{2})}{\sum_{i=t-t+1}^{t} (i-(t-t+1))(i-\frac{2t-t+1}{2})} \\
&= \frac{\sum_{i=\lfloor \frac{t+2}{2} \rfloor}^{t} (i - \frac{t+1}{2})}{\sum_{i=1}^{t} (i-1)(i-\frac{t+1}{2})} \\
&= \frac{\frac{1}{2}\left(t - \lfloor \frac{t}{2} \rfloor\right) \lfloor \frac{t}{2} \rfloor}{\frac{1}{12} t \left(t^2-1\right)} \\
&= \begin{cases} \frac{\frac{1}{8}t^2}{\frac{1}{12}(t)(t-1)(t+1)} & \text{, when } t \text{ even} \\ \frac{\frac{1}{8}(t-1)(t+1)}{\frac{1}{12}(t)(t-1)(t+1)} & \text{, when } t \text{ odd} \end{cases} \\
&= \begin{cases} \frac{3}{2}(\frac{t}{t^2-1}) & \text{, when } t \text{ even} \\ \frac{3}{2}(\frac{1}{t}) & \text{, when } t \text{ odd,} \end{cases}
\end{aligned}$$

which implies (36). □

Appendix D: Synthetic Experiment

We propose a synthetic experiment as a sensitivity analysis for how a small model misspecification could affect the final results.

Put simply, it assumes that the outcome variable and dependent variable follow a locally (dependent on $\vec{X}_{t,c}$) log-linear relation.

For our experiment, we consider a scenario where the true growth rate is locally linear instead:

$$\mathbb{E}[\boldsymbol{I}^{\boldsymbol{t}}_{\boldsymbol{s,c}} | \vec{X}_{t,c}] = \alpha(\vec{X}_{t,c}) + r(\vec{X}_{t,c})s + \mathbb{E}[\boldsymbol{\varepsilon}^{\boldsymbol{t}}_{\boldsymbol{t,c}} | \vec{X}_{t,c}] \quad \forall s \in \mathbb{Z}^+. \tag{37}$$

Experiment Setup:

Following our real life use case, we index the data observations by the tuple (t, c), where $t \in [T] := \{1, \cdots, 365\}$ is the progress of time, and $c \in C := \{1, \cdots, 1000\}$ are indexes for 1000 different counties.

At each time county pair index, the observed outcome variable is $I_{t,c} \in \mathbb{Z}_0^+$ and the observable feature variable is $\vec{X}_{t,c} \in \mathbb{R}^6$ i.e., a vector with 6 continuous features.

Let the true data generating process (DGP) for the incident case numbers be specified below:

$$I_{1,c} = 100, \forall c \in C \tag{38}$$

$$I_{t,c} = \underbrace{10\left(\sum_{k=1}^{2} X_{t,c,k}\right)}_{r(\vec{X}_{t,c})} + I_{t-1,c}, \forall t \geq 1, c \in C \tag{39}$$

In short, the true growth rate parameter is the determined by the sum of the first 2 features, with the other 4 features being irrelevant. We simulate the features and random noise as follows:

$$X_{t,c,k} \sim \text{Uniform}(0, 1), \forall t \in [T], c \in C, k \in \{1, \ldots, 6\} \tag{40}$$

$$\tag{41}$$

If our model is correctly specified as linear, then the projected forecast 7 days later from today is:

$$(I_{t+7,c}) = (I_{t,c}) + 7\hat{r}_{t,c}$$

If not correctly specified as exponential:

$$(I_{t+7,c}) = (I_{t,c}) \exp\{7\hat{r}_{t,c}\}$$

$$\ln(I_{t+7,c}) = \ln(I_{t,c}) + 7\hat{r}_{t,c}$$

To allow both models to have enough time to accumulate data for training, we start benchmarking them daily from $t = 30$ onwards.

Results:

Figures 6 and 7 present the daily MAE and RMSE of `Exponential TLRF` and `Linear TLRF` estimators, respectively. The median MAE and RMSE of these estimators are presented in Table 5. Compared to `Linear TLRF`, the misspecified exponential model estimator `Exponential TLRF` is consistently outperformed as expected.

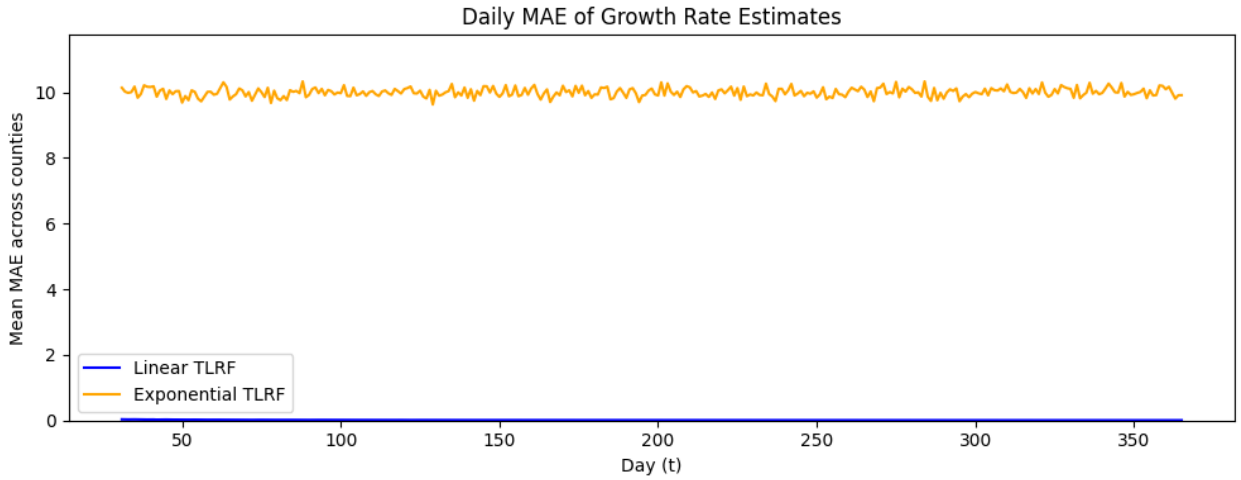

Figure 6 MAE plot of parameter estimation accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

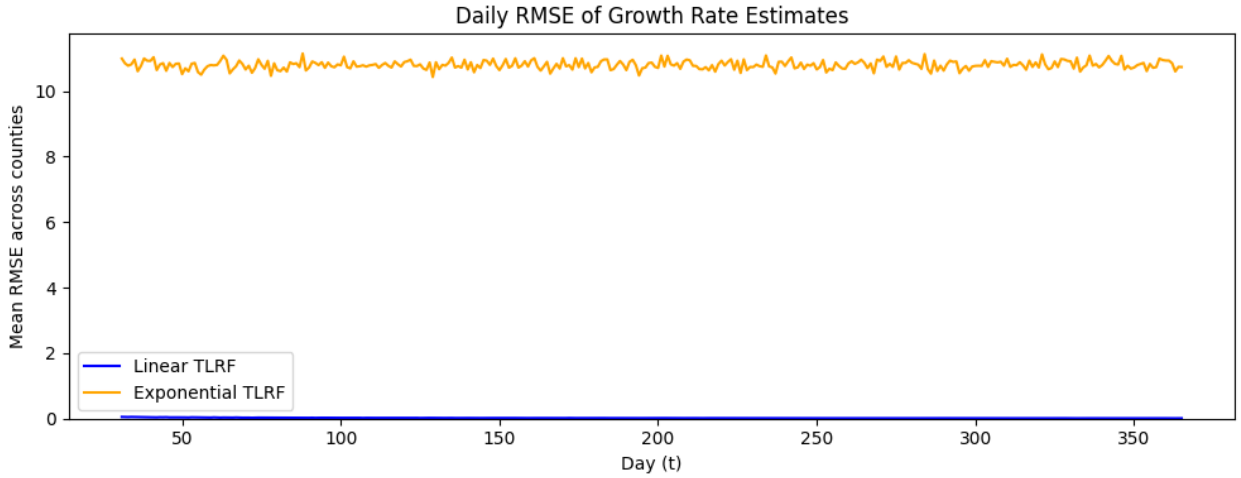

Figure 7 RMSE plot of parameter estimation accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

In addition we present the overall median of the daily MAE and RMSE of the growth rate estimates of both models in Table 5.

Subsequently, we compare the 7-day-ahead prediction accuracy between `Exponential TLRF` and `Linear TLRF`, a benchmarking methodology applied throughout the manuscript. Figures 8 and 9 present the daily MAE and RMSE of these two forecasting models, respectively. The median MAE and RMSE of both models' forecasts are presented in Table 6. Interestingly, despite parameter estimation errors, forecast accuracy

Method	MAE	RMSE
`Exponential TLRF`	10.00	10.80
`Linear TLRF`	0.013	0.018

Table 5 Median MAE and RMSE of parameter estimation accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

between `Exponential TLRF` and `Linear TLRF` remains comparable, with the gap narrowing as training data accumulates.

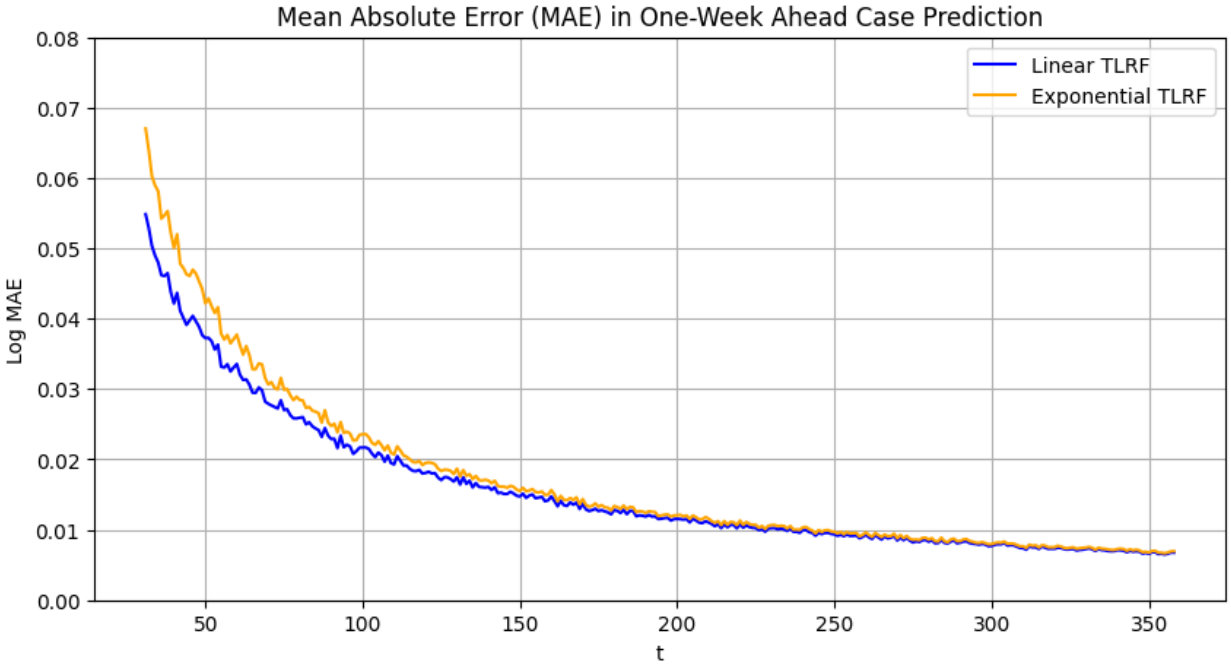

Figure 8 MAE plot of prediction accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

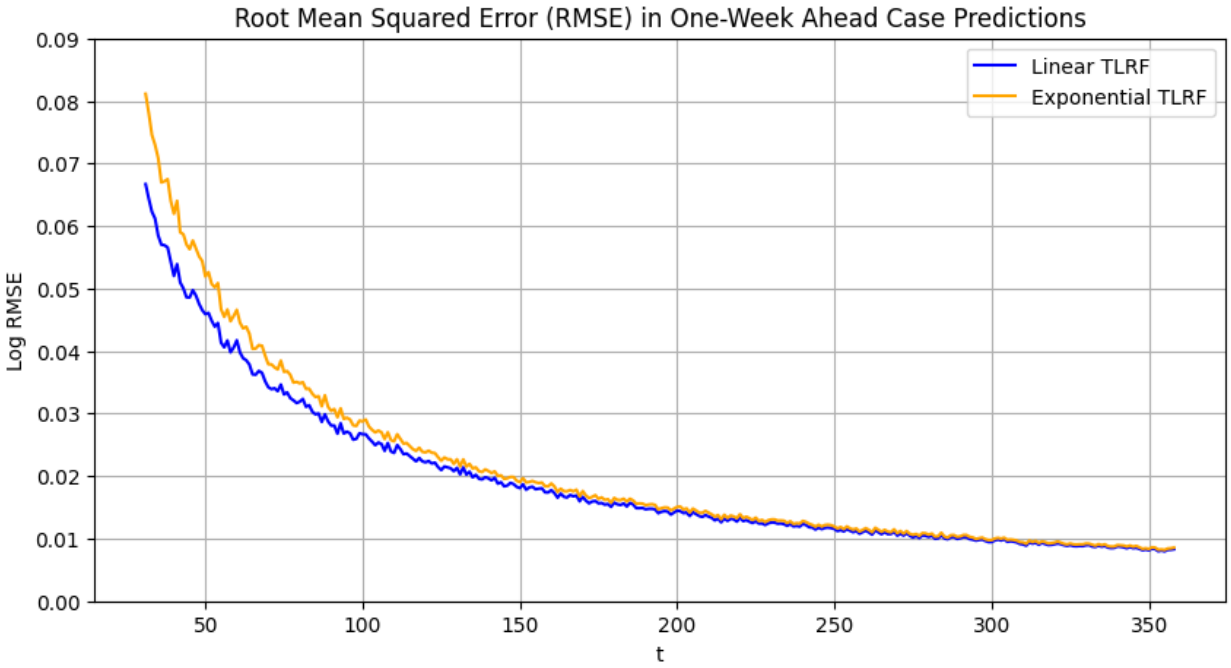

Figure 9 RMSE plot of prediction accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

Method	**MAE**	**RMSE**
`Exponential TLRF`	0.012	0.015
`Linear TLRF`	0.012	0.014

Table 6 Median MAE and RMSE of prediction accuracy (`Exponential TLRF` vs. `Linear TLRF`): Since synthetic data is generated from the locally linear model (37), the `Exponential TLRF` estimator suffers from the misspecification bias, while the `Linear TLRF` estimator is unbiased.

Experiment findings.

This synthetic experiment reveals two important insights about TLRF's behavior under model misspecification. First, TLRF exhibits a large error in parameter estimation under model misspecification, as expected. Second, surprisingly, this parameter bias translates to only modest degradation in forecast accuracy with the performance gap narrowing as training data accumulates. This demonstrates that while Assumption 1 is crucial for the correct interpretation of parameters as exponential growth rates (fundamental for epidemiological models like SIR/SEIR), TLRF maintains practical robustness for prediction tasks even under misspecification.

The code for this benchmark can be found on our GitHub repository at: https://github.com/wangzilongri/COVID-tracking/tree/TLGRF_major_revision/analysis/A1_Robustness_Check.

Appendix E: Supplementary Benchmark Implementation Details and Results

This section is dedicated to providing additional implementation details and results of the benchmarks discussed in the main text. Appendix E.1 discusses the window fitting based estimators discussed in §5. Appendix E.2 then provides further implementation details regarding the `LASSOTL` benchmark discussed in §6.

E.1. Window Based Methods

In this subsection, we provide the implementation details and additional performance metrics of the Fixed Windows estimators as referenced in §5.1, `tcv` and `ctcv` as referenced in §5.2, and the K-Means Dynamic Window procedure as described in §5.3.

E.1.1. Fixed Window Size Estimators This subsubsection contains the additional RSME plot (Figure 10) of `TLRF` against fixed window size estimators mentioned in §5.1. We can see from Figure 10 that no clear winner emerges between various fixed window size choices and that they are also uniformly outperformed by `TLRF` when evaluated upon the RMSE metric.

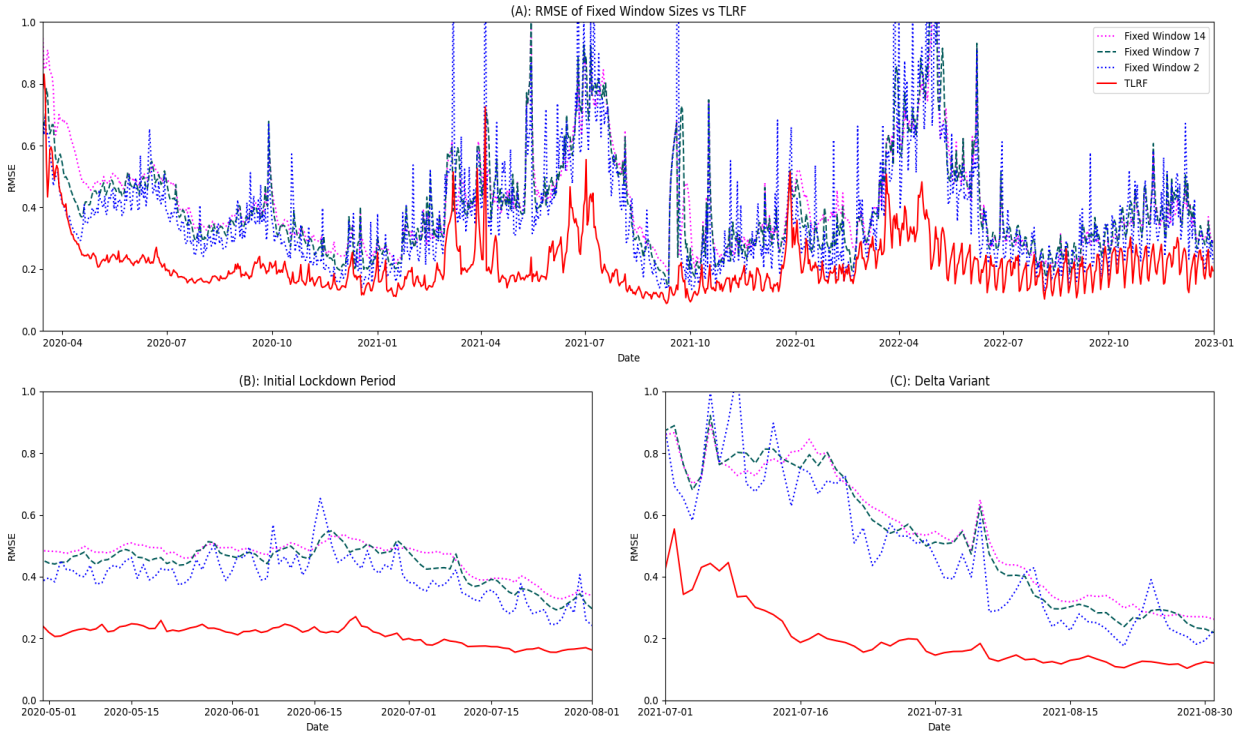

Figure 10 RMSE plot of prediction accuracy (TLRF vs. Fixed 2-, 7-, and 14-Day Fitting Window): Plot A (Top) shows the RMSE comparison across the study period. Plot B (bottom left) zooms in on the initial lockdown period. Plot C (bottom right) zooms in on the period during the outbreak of the Delta variant.

E.1.2. tcv The `tcv` estimator (19) can be directly computed from the case incidence data $\{(s, \ln(I_{s,c})) : \forall s \in \{t-\delta_t+1, \ldots, t\}, c \in C\}$ for any fitting window size hyperparameter as:

$$\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}^{\boldsymbol{ols}(\boldsymbol{\delta_t})} = \frac{\sum_{s=t-\delta_t+1}^{t} (s - \frac{\sum_{s=t-\delta_t+1}^{t} s}{\delta_t})(\ln(\boldsymbol{I_{s,c}}) - \frac{\sum_{s=t-\delta_t+1}^{t} \ln(\boldsymbol{I_{s,c}})}{\delta_t})}{\sum_{s=t-\delta_t+1}^{t} \left(s - \frac{\sum_{s=t-\delta_t+1}^{t} s}{\delta_t}\right)^2}. \tag{42}$$

This encompasses the incidence case data of every county c from the beginning of the chosen fitting window $(t-\delta_t+1)$ to the current time of query (t). As described in §5.2, the key characteristic of the `tcv` estimator is that at each time point t, all the counties are constrained to use the same fitting window size hyperparameter δ_t, i.e., the entire nation shares the same fitting window hyperparameter at every time step. We will now describe the cross-validation procedure to obtain the best δ_t, and how the final test-time estimator is computed and evaluated.

Our goal at test time is to forecast the log-incident case numbers of each county $(\ln(I_{t+7,c}))$ seven days ahead from our current time of query t. Our search range for our hyperparameter ranges from $\{2, \ldots, 14\}$. To prevent data leakage from the future, we follow the standard implementation of the rolling window time series cross validation procedure (c.f. Pedregosa et al. 2011). Specifically, the training-validation splits on

each day are generated successively to ensure that the training set is always of an earlier date than that of the validation set. This procedure is designed to be consistent with the accumulative nature of time series data to avoid leaking information from the future.

More precisely, in our validation procedure, we generate $t-20$ folds of training and validation pairs $\{(training_n, validation_n) : \forall n \in \{1,2,\ldots,t-20\}\}$, where the n-th fold consists of training data from day n to $n+13$: $training_n := \{(s, \ln(\boldsymbol{I}_{s,c})) : s \in \{n, n+1, \ldots, n+13\}, c \in C\}$ and validation ground truth of all counties on the $n+20$-th day: $validation_n := \{(s, \ln(\boldsymbol{I}_{s,c})) : s \in \{n+20\}, c \in C\}$. A detailed illustration of the training and validation data used in each fold when choosing the best δ_t is presented in Table 7.

Fold Index	Training Data	Validation Data
1	$\{(1, \ln(\boldsymbol{I}_{1,c})), \ldots, (14, \ln(\boldsymbol{I}_{14,c}))\|\forall c \in C\}$	$\{(21, \ln(\boldsymbol{I}_{21,c}))\|\forall c \in C\}$
2	$\{(2, \ln(\boldsymbol{I}_{2,c})), \ldots, (15, \ln(\boldsymbol{I}_{14,c}))\|\forall c \in C\}$	$\{(22, \ln(\boldsymbol{I}_{22,c}))\|\forall c \in C\}$
$\vdots$	$\vdots$	$\vdots$
n	$\{(n, \ln(\boldsymbol{I}_{n,c})), \ldots, (n+13, \ln(\boldsymbol{I}_{n+13,c}))\|\forall c \in C\}$	$\{(n+20, \ln(\boldsymbol{I}_{n+20,c}))\|\forall c \in C\}$
$\vdots$	$\vdots$	$\vdots$
$t-20$	$\{(t-20, \ln(\boldsymbol{I}_{t-20,c})), \ldots, (t-7, \ln(\boldsymbol{I}_{t-7,c}))\|\forall c \in C\}$	$\{(t, \ln(\boldsymbol{I}_{t,c}))\|\forall c \in C\}$

Table 7 Training and validation data of each fold for selecting hyperparameter $\delta_t \in \{2,\ldots,14\}$ for the `tcv` estimator by cross-validation

For each of the folds, we then train the `tcv` estimators using candidate window sizes $\{2,\ldots,14\}$, forecast the incident case numbers of each county in the fold, and assess the validation error of each hyperparameter choice in each fold. The validation errors for each choice of the hyperparameter are then averaged across all folds. The hyperparameter choice with the lowest validation error is assigned to δ_t for test time evaluation (forecasting $\ln(\boldsymbol{I}_{t+7,c})$). This cross validation procedure and test-time evaluation is repeated for all $\delta_t : t \in \{21,\ldots,T\}$.

E.1.3. ctcv For each county c, the `ctcv` estimator (20) can be directly computed from the case incidence data $\{(s, \ln(I_{s,c})) : \forall s \in \{t-\delta_{t,c}+1,\ldots,t\}\}$ for any fitting window size $\delta_{t,c}$ hyperparameter, in the following fashion:

$$\hat{\boldsymbol{r}}_{\boldsymbol{t,c}}^{\boldsymbol{ols}(\boldsymbol{\delta_{t,c}})} = \frac{\sum_{s=t-\delta_{t,c}+1}^{t}(s - \frac{\sum_{s=t-\delta_{t,c}+1}^{t} s}{\delta_{t,c}})(\ln(\boldsymbol{I}_{\boldsymbol{s,c}}) - \frac{\sum_{s=t-\delta_{t,c}+1}^{t} \ln(\boldsymbol{I}_{\boldsymbol{s,c}})}{\delta_{t,c}})}{\sum_{s=t-\delta_{t,c}+1}^{t}\left(s - \frac{\sum_{s=t-\delta_{t,c}+1}^{t} s}{\delta_{t,c}}\right)^2}. \tag{43}$$

This encompasses the incidence case data of the county c from the beginning of the chosen fitting window $(t-\delta_{t,c}+1)$ to the current time of query (t). In contrast to `tcv`, the key characteristic of the `ctcv` estimator

is that at each time point t, all the counties are free to use their own fitting window size hyperparameter $\delta_{t,c}$. We will now describe the cross-validation procedure to obtain the best $\delta_{t,c}$, and how the final test-time estimator is computed and evaluated.

The procedure for selecting the best hyperparameter at each time point is done independently for each county c. The test time evaluation for forecasting $\ln(\boldsymbol{I}_{t+7,c})$ is consistent with `tcv` as described in Appendix E.1.2. Therefore, for each county c at time point t, we will generate $t-20$ folds of training and validation pairs as per Table 8.

Fold Index	Training Data	Validation Data
1	$\{(1,\ln(\boldsymbol{I}_{1,c})),\ldots,(14,\ln(\boldsymbol{I}_{14,c}))\}$	$\{(21,\ln(\boldsymbol{I}_{21,c}))\}$
2	$\{(2,\ln(\boldsymbol{I}_{2,c})),\ldots,(15,\ln(\boldsymbol{I}_{14,c}))\}$	$\{(22,\ln(\boldsymbol{I}_{22,c}))\}$
$\vdots$	$\vdots$	$\vdots$
n	$\{(n,\ln(\boldsymbol{I}_{n,c})),\ldots,(n+13,\ln(\boldsymbol{I}_{n+13,c}))\}$	$\{(n+20,\ln(\boldsymbol{I}_{n+20,c}))\}$
$\vdots$	$\vdots$	$\vdots$
$t-20$	$\{(t-20,\ln(\boldsymbol{I}_{t-20,c})),\ldots,(t-7,\ln(\boldsymbol{I}_{t-7,c}))\}$	$\{(t,\ln(\boldsymbol{I}_{t,c}))\}$

Table 8 Training and validation data of each fold for selecting hyperparameter $\delta_{t,c} \in \{2,\ldots,14\}$ for the `ctcv` estimator by cross-validation

The hyperparameter selection process, followed by the test time evaluation (forecasting $\ln(\boldsymbol{I}_{t+7,c})$), will proceed similarly to the rolling window time series cross-validation procedure described in Appendix E.1.2. The only difference is that each county c will now choose its own best $\delta_{t,c}$ independently.

E.1.4. K-Means Dynamic Window The K-Means Dynamic Window method obtains its estimator in two steps. First, utilizing spatial information (e.g., HSA regions, Census information, Proportion of vulnerable populations), it conducts K-Means clustering on all counties in the U.S., where K is the total number of clusters. Second, once the K clusters $\{C_1, C_2, \ldots, C_K\}$ have been generated, it chooses a cluster-specific fitting window for counties within the same cluster via time-based cross-validation.

We will first describe the clustering process in detail. We then describe the hyperparameter selection process for the dynamic window estimators within a given cluster.

Clustering Implementation

The goal of the clustering step is to partition all the counties of the US into $K \in \{100, 200, \ldots,\}$ clusters using geographic (i.e., time-invariant) information from our datasets. We first determined at the dataset level what features are time-invariant within our study:

1. The 2019 United States Census Gazetteer Files (c.f. United States Census Bureau 2019). Sample features included are:

- Geographic Centers (Latitude, Longitude) of Counties

2. The Centers for Disease Control and Prevention Social Vulnerability Index 2018 Database (c.f. Centers for Disease Control and Prevention 2018). Sample features included are:
   - Population below the poverty line
   - Unemployment Rate
   - Proportion of Elderly (> 65 years of age)
3. The CDC SARS-CoV2 Variant Proportions Dataset (c.f. Centers for Disease Control and Prevention 2021a). We only utilize the information of which county belongs to which Health and Human Services (HHS) Region.

The K clusters are then randomly initialized using the `k-means++` procedure by Arthur and Vassilvitskii (2007), where the clustering procedure was run for at most $10,000$ iterations.

This parallel implementation can be found on our GitHub repo: https://github.com/wangzilongri/COVID-tracking/tree/TLGRF_major_revision/analysis/benchmark_tcv_kmeans_code

Dynamic Window Selection per Cluster

For a given choice of K, once we have obtained the K clusters $C_1, \ldots, C_K$, we can proceed with fitting the dynamic window estimator within each cluster in a similar fashion to the previously described estimators `tcv`,`ctcv`. For each county c in the k-th cluster C_k, at time point t, the K-Means Dynamic Window estimator (21) can be directly computed from the case incidence data $\{(s, \ln(I_{s,c})) : \forall s \in \{t-\delta_{t,k}+1, \ldots, t\}, \forall c \in C_k\}$ for any fitting window size hyperparameter $\delta_{t,k}$, in the following fashion:

$$\hat{\boldsymbol{r}}_{\boldsymbol{t,k}}^{\boldsymbol{ols}(\boldsymbol{\delta_{t,k}})} = \frac{\sum_{s=t-\delta_{t,k}+1}^{t} (s - \frac{\sum_{s=t-\delta_{t,k}+1}^{t} s}{\delta_{t,k}})(\ln(\boldsymbol{I_{s,c}}) - \frac{\sum_{s=t-\delta_{t,k}+1}^{t} \ln(\boldsymbol{I_{s,c}})}{\delta_{t,k}})}{\sum_{s=t-\delta_{t,k}+1}^{t} \left(s - \frac{\sum_{s=t-\delta_{t,k}+1}^{t} s}{\delta_{t,k}}\right)^2}. \tag{44}$$

The procedure for selecting the best hyperparameter at each time point is done independently for each cluster C_k. The test time evaluation for forecasting $\ln(\boldsymbol{I}_{t+7,c})$ is consistent with `tcv` and `ctcv` as described in Appendix E.1.2 and Appendix E.1.3 respectively. Each cluster C_k independently selects its best hyperparameter $\delta_{t,k}$ at every time point t, which will be shared for all counties $c \in C_k$. Therefore, at time point t, for each cluster C_k, we will generate $t-20$ folds of training and validation pairs as per Table 9. Once each cluster obtains its hyperparameter choices from the rolling window time series cross-validation procedure, we will proceed with the test time evaluation of forecasting $\ln(\boldsymbol{I}_{t+7,c})$ for all time points and all counties across all clusters.

Notably, this estimator reduces to the `tcv` estimator (19) if its fitting window size is the same for all U.S. counties each day t, i.e., $\delta_{t,k} = \delta_t$, by assigning all counties to the same cluster. In contrast, this estimator becomes the `ctcv` estimator (20) if each county is assigned to its own distinct cluster, i.e., $\delta_{t,k} = \delta_{t,c}$.

The parallel implementation of this procedure can be found on our GitHub repository at: https://github.com/wangzilongri/COVID-tracking/tree/TLGRF_major_revision/analysis/benchmark_tcv_kmeans_code/cumsum_code.

Fold Index	Training Data	Validation Data
1	$\{(1,\ln(\boldsymbol{I}_{1,c})),\ldots,(14,\ln(\boldsymbol{I}_{14,c}))\|\forall c\in C_k\}$	$\{(21,\ln(\boldsymbol{I}_{21,c}))\|\forall c\in C_k\}$
2	$\{(2,\ln(\boldsymbol{I}_{2,c})),\ldots,(15,\ln(\boldsymbol{I}_{14,c}))\|\forall c\in C_k\}$	$\{(22,\ln(\boldsymbol{I}_{22,c}))\|\forall c\in C_k\}$
$\vdots$	$\vdots$	$\vdots$
n	$\{(n,\ln(\boldsymbol{I}_{n,c})),\ldots,(n+13,\ln(\boldsymbol{I}_{n+13,c}))\|\forall c\in C_k\}$	$\{(n+20,\ln(\boldsymbol{I}_{n+20,c}))\|\forall c\in C_k\}$
$\vdots$	$\vdots$	$\vdots$
$t-20$	$\{(t-20,\ln(\boldsymbol{I}_{t-20,c})),\ldots,(t-7,\ln(\boldsymbol{I}_{t-7,c}))\|\forall c\in C_k\}$	$\{(t,\ln(\boldsymbol{I}_{t,c}))\|\forall c\in C_k\}$

Table 9 Training and validation data of each fold for selecting hyperparameter $\delta_{t,k}\in\{2,\ldots,14\}$ for all counties in the k-th cluster at time t

E.2. LASSO Based Transfer Learner

In this subsection, we provide the implementation details of the `LASSOTL` benchmark derived from Bastani (2021) along with additional performance benchmarks in comparison to `TLRF`.

We utilize the identification result from Corollary (1): where the county-level instantaneous exponential growth rate $r_{t,c}$ is identified as

$$r_{t,c}=\mathbb{E}[\ln(\boldsymbol{I_{t,c}})-\ln(\boldsymbol{I_{t-1,c}})|\vec{X}_{t,c}].$$

As such, by denoting

$$Y_{gold}=\ln(I_{t,c})-\ln(I_{t-1,c}),\quad \vec{X}_{gold}=\vec{X}_{t,c},$$

and

$$\vec{Y}_{proxy}=[\ln(I_{t',c'})-\ln(I_{t'-1,c'})]_{(t',c')\neq(t,c)},\quad \mathcal{X}_{proxy}=[\vec{X}_{t',c'}]_{(t',c')\neq(t,c)},$$

we implemented the `LASSOTL` estimator in two stages:

Step 1 : given regularization penalty parameter μ,

$$\hat{\vec{\beta}}_{proxy}=\arg\min_{\vec{\beta}}\left\{\frac{1}{t|C|-1}\left\|\vec{Y}_{proxy}-\mathcal{X}_{proxy}^{T}\vec{\beta}\right\|_2^2+\mu\|\vec{\beta}\|_1\right\}$$

Step 2 : given regularization penalty parameter λ,

$$\hat{\vec{\beta}}_{joint}=\arg\min_{\vec{\beta}}\left\{\left\|Y_{gold}-\vec{X}_{gold}^{T}\vec{\beta}\right\|_2^2+\lambda\left\|\vec{\beta}-\hat{\vec{\beta}}_{proxy}\right\|_1\right\},$$

and obtain the `LASSOTL` estimator for the exponential growth rate:

$$\hat{r}_{t,c}^{LASSOTL}=\vec{X}_{gold}^{T}\hat{\vec{\beta}}_{joint}.$$

This was then used to generate the 7 days ahead prediction of the log incident cases $\ln(\hat{I}_{t,c}^{LASSOTL})$ and evaluated for test-time accuracy via the RMSE and MAE metric to remain consistent with our benchmarking approaches.

For each query time t, our approach involves one Step 1 estimator and $|C|$ separate Step 2 estimators, where C represents the set of counties or subgroups being studied. The hyperparameters tuned were the LASSO penalties μ for Step 1 and λ for Step 2. To select the optimal hyperparameters, we performed grid searches over the range $\{10^{-5}, 10^{-4}, \ldots, 10^{0}, 10^{1}, 10^{2}\}$. The tuning process was carried out independently for each step using 5-fold cross-validation to ensure robust performance. Both Step 1 and Step 2 were implemented using the `Python` library `scikit-learn` (Pedregosa et al. 2011), leveraging its `Lasso` and `GridSearchCV` functionalities for model fitting and hyperparameter optimization.

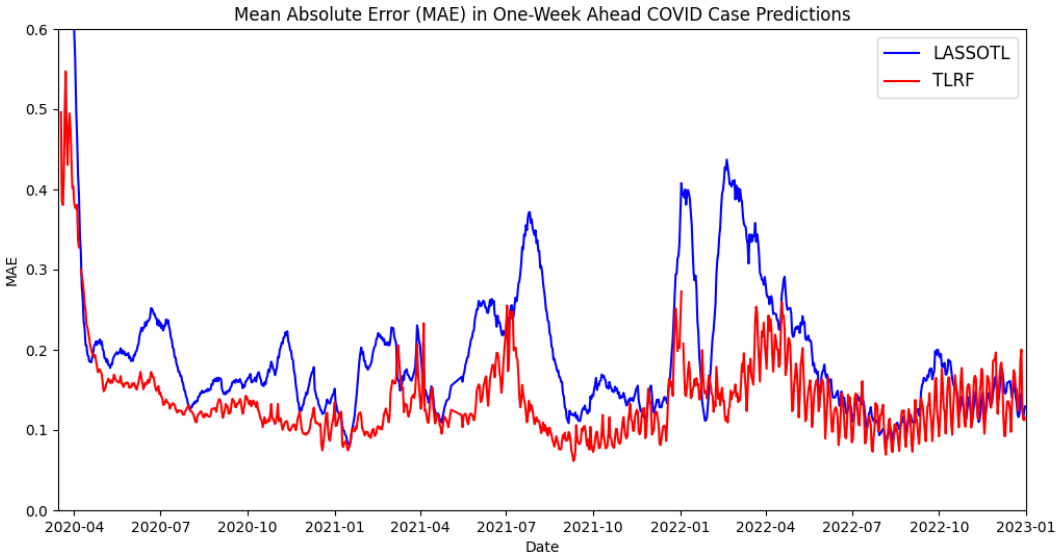

Figure 11 MAE plot of prediction accuracy (`TLRF` v.s. `LASSOTL`)

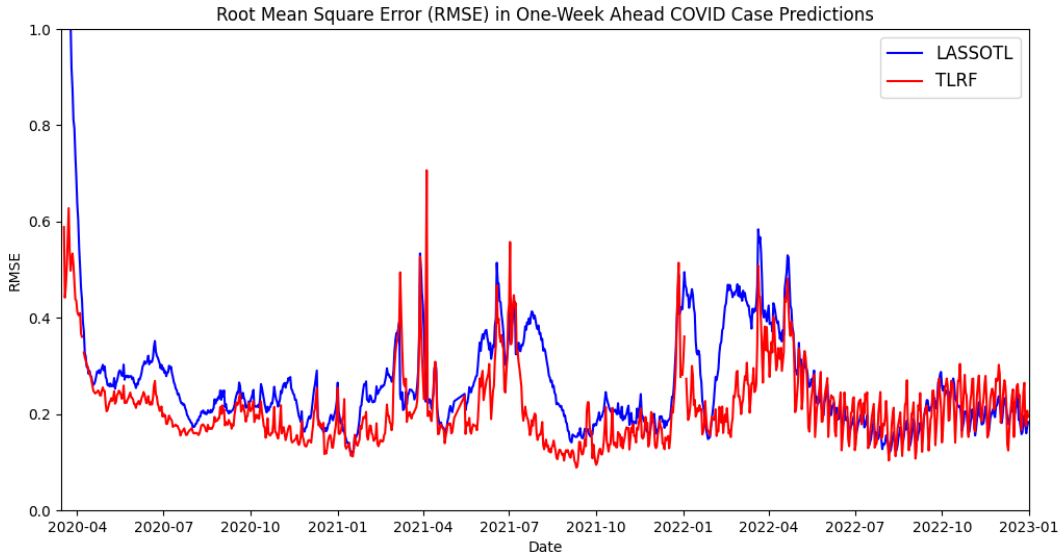

Figure 12 RMSE plot of prediction accuracy (TLRF v.s. LASSOTL)

We also present supplementary results in the form of daily MAE and RMSE plots in Figures 11 and 12 respectively, which demonstrate that TLRF outperforms LASSOTL for this use case.

The code for this benchmark can be found on our GitHub repository at: https://github.com/wangzilongri/COVID-tracking/tree/TLGRF_major_revision/analysis/benchmark_transfer_learning.

Appendix F: Details Regarding Dataset

This section provides an overview of the dataset used by TLRF in our study. First, in Appendix F.1, we describe the various datasets employed in our study. Next, in Appendix F.2, we explain the methodology used to define the incident case numbers, which serve as the dependent variable in our task. Finally, in Appendix F.3, we introduce the concept of feature importance and discuss the key features identified by TLRF.

F.1. Feature Data

All feature data we used, i.e., $\{\vec{X}_{t,c}\}_{t\in\mathbb{Z}^+,c\in C}$, are publicly available online. In this section, we briefly describe what these datasets are and how we incorporated them in our work.

F.1.1. 2019 US Census Gazetteer Files The 2019 United States Census Gazetteer Files (c.f. United States Census Bureau 2019) were used to obtain the geographic locations (latitude-longitude centroids) of each county officially registered by the United States Census Bureau. This provides a spatial feature space for the random forest algorithm to further split upon.

F.1.2. Centers for Disease Control and Prevention Social Vulnerability Index 2018 Database The Social Vulnerability Index (SVI) database (c.f. Centers for Disease Control and Prevention 2018) is a compilation of socio-economic factors such as unemployment rate, poverty rate, education attainment level etc. at the county level. These features are also included in our methodology.

F.1.3. COVID-19 US state policy database (CUSP) The CUSP database (c.f. Julia 2020) tracks when each state implemented and ended policies such as mask mandates, lockdowns, economic policies in response to the COVID-19 pandemic. As such, it is a vector of features capturing daily policy changes in each state. As these policies is state-wide, we naturally extend them to the respective county level.

F.1.4. The COVID Tracking Project From the COVID Tracking Project (c.f. The Atlantic 2018), we obtained the daily numbers of PCR, Antibody and Antigen tests performed and their positivity rates at each state. These are used as features in our estimation algorithm as well.

F.1.5. SARS-CoV2 Variant Proportions This dataset provided by the CDC (c.f. Centers for Disease Control and Prevention 2021b) contains the mean estimate (along with its lower and upper estimation bounds) of the share of variants, e.g., "B.1", "P.1", "XBB.1.5", "others" of each Health and Human Services (HHS) region every week (or biweekly if indicated) starting from the 2nd of January 2021. Each date and region record can potentially have multiple published estimates, and we decided to take the latest published estimates (not later than the day of estimation). For data imputation, each county can be mapped to its HHS region, and for the days in between every reporting interval, we decided to forward fill the missing entries (i.e., for the days between week T to week $T+1$, we will use the entries from week T). For the dates beyond the last reporting date, we also decided to forward fill. For the dates before the first record, we decided to backfill, i.e., for dates before the 2nd of January 2021, we will use the entries for the 2nd of January 2021. The variants' proportions are then normalized into percentages and joined with our current dataset to be used as features for TLRF

F.2. COVID-19 Case Incidence

In epidemiology, case incidence, or incident case number, measures the number of new occurrences of a disease in a population over a specified period of time (McNutt and Krug 2016). More precisely, the incident case number at county c on day t is defined as $I_{t,c} := C_{t,c} - C_{t-\Delta t,c}$ for $t > \Delta t$, where $C_{t,c}$ and $C_{t-\Delta t,c}$ are the cumulative case number at county c up to day t and $t - \Delta t$, respectively. Case incidence describes how fast a disease spreads in a population, and thus is commonly used to guide policy decisions such as school closures (Kittitas County 2020).

Another useful measure in epidemiology is the active case number, which captures the number of still "infectious" cases. More precisely, the active case number at county c on day t is defined as

$$A_{t,c} := C_{t,c} - D_{t,c} - R_{t,c}, \tag{45}$$

where $C_{t,c}$, $D_{t,c}$ and $R_{t,c}$ are the cumulative number of cases , deaths and recoveries at county c up to day t. Clearly, calculating the active case number requires more information than calculating the incident case number, and thus may not always be feasible. In fact, we are unable to directly calculate the number of active cases as in (45), because the number of recovered cases R_t are not reported at county-level in the U.S.. In addition, we find that the cumulative number of case and death numbers are not always consistent in our dataset (c.f. The New York Times 2020), i.e., $\exists c \in C, t \in \mathbb{Z}^+ \; s.t. \; C_{t,c} < D_{t,c}$.

To better capture COVID-19 case growth trajectories, we focus on analyzing the 22-day incident case number in this paper, i.e.

$$I_{t,c} := \begin{cases} C_{t,c} - C_{t-22,c} & \text{, if } t \geq 23 \\ C_{t,c} & \text{, if } 1 \leq t < 23. \end{cases} \tag{46}$$

Here, defining the incident case using a period of 22 days was based on the early practices observed during the pandemic. Specifically, patients were recommended to undergo a self-quarantine period of 17-22 days to confirm recovery from COVID-19 (c.f. Zhang 2021). Therefore, we chose the most stringent duration of 22 days to account for potential delays in resolving active cases. Notably, other studies also use a similar time to resolution of COVID-19 (c.f. Ben Phillips 2020). As such, (46) measures how quickly COVID-19 spreads in a county. In addition, (46) provides a good proxy for the active case number (45) as well, assuming that a COVID-19 case takes at most 22 days to resolve. Lastly, we use the 7-day moving average to smooth out oscillations in case number reporting (c.f. Bergman et al. 2020) and exclude $I_{t,c}$ observations with value below 20 in this study.

F.3. Feature Importance

F.3.1. Feature Importance by Splitting Frequency and Depth The native `variable_importance` method of the `grf` package calculates the importance of features by tallying how often the individual trees split upon said feature weighted by how early they are split within each tree. The earlier the split, the more informative the feature is and hence it holds higher weight. More formally, if we wish to compute the feature importance of feature f, for each tree in the grf, we traverse the tree and check each node to see if said node split on f. The level of the node increments by 1 as we descend the tree, with the root node having a level of 0. For every level we descend, we halve the importance of the contribution of the feature. This can be succinctly represented in equation (47),

$$\texttt{Importance}(f) = \sum_{b \in B} \sum_{node \in b} \mathbb{1}\{\texttt{node split on } f\}^{-2 \times \texttt{level(node)}} \tag{47}$$

where B is a forest consisting of tress b; each tree b consists of nodes (representing splits); the level of the node is given by `level(node)`.

F.3.2. Top 20 Most Important Features We show the Top 20 features which we presented in Appendix F.3.3 and explain what each feature means in descending order of importance.

For the purposes of our algorithm, if the feature is a date, we transformed it into a vector representing how many days have elapsed since the policy was implemented, with all days before it being 0, and the start date itself being 1. For example, if a county c's feature f date is $2020-02-05$ i.e., policy f was implemented for that county on the 5th of May 2020, the feature for county c would look like:

$$\left[0, \ldots, 0, \underbrace{1}_{2020-02-05}, 2, 3, \ldots\right]$$

Top 20 Features in Descending Order of Importance

1. Stopped Personal Visitation in State Prisons
   - The date a state stopped personal visitations to state prisons
2. State of emergency issued
   - The date a state first issued any type of emergency declaration
3. Expanding Supplemental Nutrition Assistance Program (SNAP)
   - The date a state was approved the use of a waiver to provide many SNAP households with emergency supplementary benefits up to the maximum benefit a household can receive
4. Closed K-12 Public Schools
   - The date states closed K-12 public schools
5. Closed Bars
   - The date states closed bars statewide. Unless otherwise noted, bars are defined as establishments that derive more than 50 percent of gross revenue from the sales of alcoholic beverages
6. Expanding Medicaid benefits (1135 Waivers)
   - The date a state used a 1135 waiver to modify or waive Medicaid requirements
7. Closed Gyms
   - The date states closed gyms statewide.
8. Closed Restaurants Except Takeout
   - The date when restaurants are closed for in person dining with the exception of takeout orders
9. Closed Other Non-Essential Businesses
   - The date a state closed non-essential businesses statewide. Only included directives/orders
10. Allow/Expand Medicaid Telehealth Coverage
    - The date a state expanded Telehealth coverge for Medicaid recipients
11. Closed Movie Theaters
    - The date a state closed cinemas and theaters
12. Variant - B.1.617.2

- Proportion of positive COVID test cases being of the ‘B.1.617.2’ strain, updated bi-weekly by the CDC

13. Ratio of Positive Tests
    - Ratio of Positive COVID-19 tests across a rolling 7 day average
14. Variant - Other
    - Proportion of positive COVID test cases with undetermined/unrecorded strains, udpated bi-weekly by the CDC
15. Variant - AY.3
    - Proportion of positive COVID test cases being of the ‘AY.3’ strain, updated bi-weekly by the CDC
16. Date General Public Became Eligible for COVID-19 Vaccination
    - The date in which a state made the general public eligible for COVID-19 vaccination.
17. Date Adults Ages 30+ Became Eligible for Covid-19 Vaccination
    - The date in which a state made adults ages 30+ eligible for COVID-19 vaccination
18. Allowed Restaurants to Sell Takeout Alcohol
    - The date when restaurants, not classified as bars by percentage of revenue, are allowed to sell takeout alcohol
19. Variant - BA.1.1
    - Proportion of positive COVID test cases being of the ‘BA.1.1’ strain, updated bi-weekly by the CDC
20. Date Adults Ages 80+ Became Eligible for Covid-19 Vaccination
    - The date in which a state made adults ages 80+ eligible for COVID-19 vaccination

F.3.3. Feature Importance and Transfer Learning As shown in Appendix A, TLRF derives much of its predictive power from transfer learning. This finding suggests that instead of relying on possibly limited historical data of a county in isolation for training, we can pool together data of counties which experienced similar events to generate a more robust and accurate rate estimate. A remaining question though is “what are the most important determinants of ‘similarity’ across counties”. Motivated by this question, in this section, we analyze the most important features used by TLRF in determining similarity across time and space.

We discovered that TLRF assigns non-zero importance to 398 out of the 468 of features incorporated in our dataset. In other words, our algorithm uses at least 398 features to determine which proxy data should be pooled together with the target data for estimation purposes.

Figure 13 presents the top 20 most important features used by TLRF in determining similarity in terms of COVID-19 spreading patterns. The top 20 most important features account for over 66.6% of the total feature importance. Amongst them, we first observe that policies that aim to reduce social interactions (e.g., closing of bars, closing of restaurants, closing of non-essential businesses, stopping personal visitation to

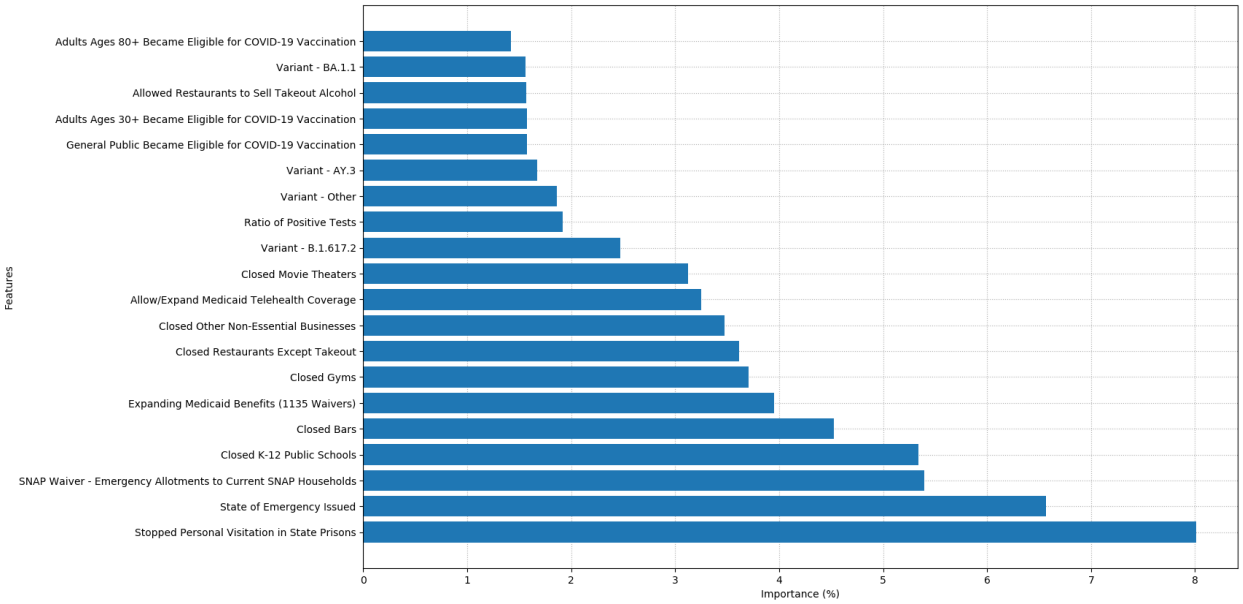

Figure 13 Top 20 most important features of TLRF based on weighted split frequency

prisons, state of emergency issued, closing of K12 public schools etc.) collectively account for a total of 43.7 out of 100 percent of the weighted splits. Next, safety net policies (Expanding Supplementary Nutrition Benefits, Expanding Medicaid Benefits w/ 1135 Waiver, Expanding Teleheath Coverage) account for a total of 12.6%. We then see that the proportion of positive COVID-19 tests and various strains account for 9.49%. Finally, we see that policies pertaining to the availability of vaccinations account for a total of 4.6%. These findings indicate that isolation policies, safety net policies, COVID-19 strain information, and vaccination availability provide the strongest signals in determining experienced growth rates and county similarity for our algorithm.

Appendix G: Additional Analyses for the CDPHE Case Study

In this appendix subsection, we present additional information complementing our case study in §7. First, in Appendix G.1, we provide further details regarding the number of decision points and the counties investigated. Following that, in Appendix G.2, we conduct a more in-depth analysis comparing the performance of TLRF and CDPHE. Then, in Appendix G.3, we consider an alternative use case of TLRF by considering a threshold based policy for allocating investigation resources. Finally, in Appendix G.4, we consider plugging in our instantaneous growth rate estimates from TLRF into a compartmental model and show that it can be used to improve incident case estimates in Colorado.

G.1. Decision Points and Outbreak Investigations

Our study period, which spans 2020-03-17 to 2022-12-31, has a total of 159 decision points (DPs) and 182

investigated counties. A plot of how many investigations were made at each date is given below in Figure 14 and in Table 10.

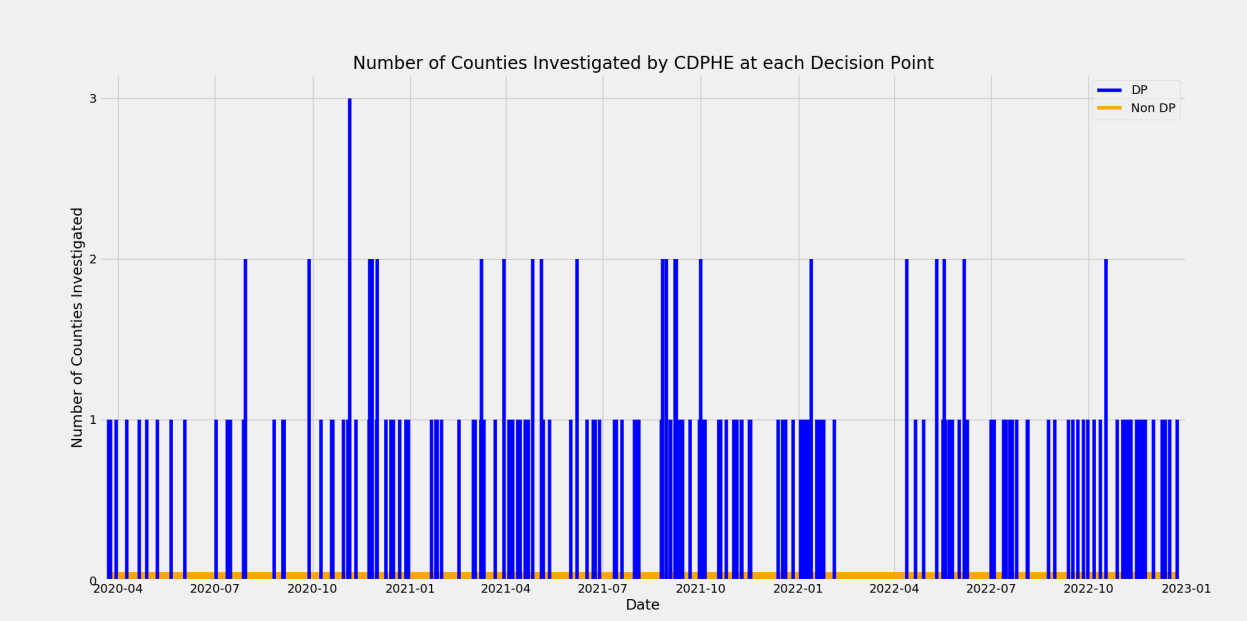

Figure 14 Daily Number of New Investigations Initiated by CDPHE

	# of Days	Daily # of New Investigations
Non-decision points	861	0
Decision points	137	1
	21	2
	1	3

Table 10 Distribution of Daily Number of New Investigations Initiated by CDPHE

G.2. TLRF vs. CDPHE Further Analyses

We further examine if there are systematic differences in counties chosen for investigations by CDPHE and TLRF. We also investigate what types of counties are most frequently misclassified by the CDPHE and TLRF, respectively.

First, we compare CDPHE to TLRF's performance in the more populous counties in Table 11. We observe that both the CDPHE and TLRF exhibit higher PPV and F1 Scores in this analysis when we focus on the more populous subset of counties in Colorado. However, TLRF still consistently outperforms CDPHE across

all subsets and metrics. We provide an intuitive explanation behind this. As we explained §7, the TLRF decision criterion is implicitly weighted by the population of counties as it prioritizes counties with high current reported incident case numbers. Therefore, the TLRF algorithm is inherently capturing CDPHE's investigation prioritization of outbreaks in more populous counties, but in a more targeted manner.

Most Populous Counties	CDPHE		TLRF	
	PPV	F1	PPV	F1
Top 10%	0.3333	0.3636	1.0000	1.0000
Top 20%	0.3571	0.2020	0.6974	0.7211
Top 50%	0.2185	0.2023	0.6268	0.6357
All	0.1813	0.1813	0.5879	0.5879

Table 11 Positive Predictive Value (PPV) and F1 Score of CDPHE and TLRF on the more populous counties of Colorado

Finally, Tables 12 and 13 present the summary statistics of counties most frequently misclassified by CDPHE and TLRF, respectively. Notably, our investigation reveals significant overlap in the commonly misclassified counties by CDPHE and TLRF. Specifically, the most frequently misclassified counties, on average, exhibit a lower total population size and other notable census statistics (e.g., individuals facing poverty, unemployment, or disabilities) in comparison to the state average. These findings suggest that CDPHE and TLRF share similar empirical patterns in terms of their investigation priorities.

Counties of:	Statistics	Total Population	Poverty	Unemployed	Age > 65	Age < 17	Disabled	Uninsured
CDPHE Incorrect	Mean	35495.4	3572.0	839.3	5491.2	7386.2	3635.9	3265.0
	Median	18325.0	2084.0	531.0	3459.0	3381.0	2332.0	1864.0
CDPHE Correct	Mean	69929.6	8138.9	1602.4	9320.1	14571.1	7039.1	7057.1
	Median	27754.0	3296.0	736.0	5076.0	5845.0	2990.0	2414.5
All Colorado	Mean	43319.7	4568.9	1037.7	6468.0	9004.7	4379.4	4226.2
	Median	25162.0	3075.0	589.0	3689.0	5514.0	2939.0	2395.0

Table 12 Comparison of Mean and Median statistics of the Top 10 most frequently misclassified and correctly classified Colorado counties by CDPHE

Counties of:	Statistics	Total Population	Poverty	Unemployed	Age > 65	Age < 17	Disabled	Uninsured
TLRF Incorrect	Mean	25699.0	2447.4	601.3	4073.5	5181.0	2570.6	2606.6
	Median	17909.0	1890.0	500.0	3253.0	2954.0	2332.0	1762.0
TLRF Correct	Mean	58924.5	6003.8	1338.8	8490.7	11784.9	5618.5	5637.7
	Median	55101.0	4609.0	1250.0	8634.0	10565.0	4854.0	4993.0
All Colorado	Mean	43319.7	4568.9	1037.7	6468.0	9004.7	4379.4	4226.2
	Median	25162.0	3075.0	589.0	3689.0	5514.0	2939.0	2395.0

Table 13 Comparison of Mean and Median statistics of the Top 10 most frequently misclassified and correctly classified Colorado counties by TLRF

G.3. A Threshold Policy Derived from TLRF

This section delves into an alternative use case of TLRF, building on the use case discussed in §7. Specifically, TLRF can be utilized to configure an absolute threshold policy. That is, instead of choosing a relative threshold as in §7, we can select a fixed threshold. This absolute threshold policy would then recommend investigating all counties that have caseload increases beyond this threshold projected by TLRF. In other words, the less stringent or lower this predetermined threshold is, the more counties would be investigated under this policy. To quantify the number of investigations recommended by absolute threshold policies, we have experimented with different thresholds and plotted two of them in Figure 15.

We now discuss the practical implications of implementing policies based on absolute thresholds. As shown in Figure 15, it is difficult to integrate CDPHE's investigation resource constraint into absolute threshold-based policies. On one hand, when a lower threshold is chosen (e.g., Threshold=10), this policy would recommend a large daily investigation number that exceeds the capacity constraint of CDPHE. On the other hand, when a higher threshold is chosen (e.g., Threshold=30), this policy would recommend too few investigations on some days when investigation resources are available. In general, it can be challenging for an absolute threshold policy to match the investigation frequency of CDPHE. Therefore, recommendations made by an absolute threshold policy may not always be feasible for CDPHE.

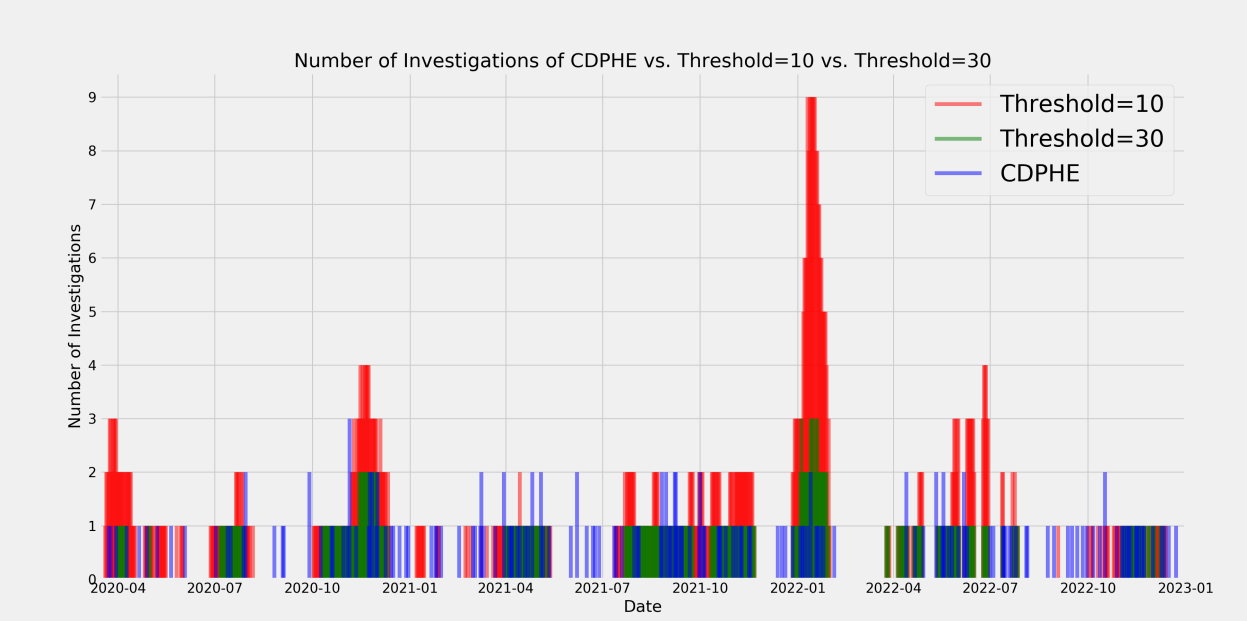

Figure 15 Daily Number of New Investigations Initiated by CDPHE

vs.

Daily Number of New Investigations Recommended by the Absolute Threshold Policy with Threshold=10 and Threshold=30

G.4. Using TLRF to Improve SEIR Estimates of Colorado Counties

In this subsection, we integrate our growth rate estimates into established epidemiological models like that of the SEIR framework.

We implemented an SEIR model with the following structure:

$$\begin{aligned}\frac{dS}{dt} &= -\frac{\beta SI}{N}\\ \frac{dE}{dt} &= \frac{\beta SI}{N} - \sigma E\\ \frac{dI}{dt} &= \sigma E - \gamma I\\ \frac{dR}{dt} &= \gamma I,\end{aligned} \tag{48}$$

where S, E, I, and R represent Susceptible, Exposed, Infectious, and Recovered populations, respectively, and N represents the total population, which is the sum of the previous four terms. Following Ma (2020), the exponential growth rate r relates to model parameters of (48) through

$$r = \frac{-(\sigma+\gamma) + \sqrt{(\sigma-\gamma)^2 + 4\sigma\beta}}{2}, \tag{49}$$

which implies that

$$\beta = \frac{(2r + (\sigma+\gamma))^2 - (\sigma-\gamma)^2}{4\sigma} \tag{50}$$

where β is the transmission rate; σ is the incubation rate; γ is the recovery rate. This relation allows us to incorporate our TLRF growth rate estimates into the SEIR framework. As we have independently estimated the exponential growth rate $r_{t,c}$ for each time t and county c. Equation (50) allows us to reduce the free parameters to σ and γ, which we estimate non-parametrically through a data-driven method described below.

In this setup, the parameters $(\beta_{t,c}, \sigma_{t,c}, \gamma_{t,c})$ are allowed to vary over time and across counties, with $\beta_{t,c}$ being derived from Equation (50). To manage computational costs, we constrain $(\sigma_{t,c}, \gamma_{t,c})$ to remain constant within each county c for a given calendar month. We search for the optimal values of $\sigma_{t,c}$ and $\gamma_{t,c}$ from the set $\{0.01, 0.02, \ldots, 0.99, 1.0\}$, while allowing $\beta_{t,c}$ to vary on a daily basis.

Our benchmark criterion remains consistent: we aim to forecast the log of incident case numbers, $\ln(I_{t+7,c})$, seven days ahead for counties in Colorado. The study period spans from April 1, 2020, to December 31, 2022.

For each county in Colorado, we generate a separate SEIR trajectory to reflect the local dynamics of disease spread. To avoid data leakage from future time points, we use an iterative procedure based on retrospective data to select the optimal parameters $(\sigma_{t,c}, \gamma_{t,c})$:

1. **Validation Phase:** For month T, we search for the best values of $(\sigma_{t,c}, \gamma_{t,c})$ that minimize the root mean squared error (RMSE) between the SEIR model's daily incident case numbers $I_{t,c}$ and the actual observed values.
2. **Prediction Phase:** To make forecasts for month $T + 1$, we fix the parameter estimates $(\sigma_{t,c}, \gamma_{t,c})$ using the best-performing values from the validation phase in month T. At each time point t, we use the current state $(S_{t,c}, E_{t,c}, I_{t,c}, R_{t,c})$ and the fixed parameters $(\beta_{t,c}, \sigma_{t,c}, \gamma_{t,c})$ to generate the SEIR trajectory 7 days ahead. We then compare the predicted $\ln(I_{t,c})$ with the actual values using RMSE and MAE error metrics. This procedure is repeated for each subsequent month, leveraging temporal consistency while allowing for periodic adjustments to reflect evolving epidemiological conditions.

To evaluate the impact of our method on SEIR model disease forecasting, we compare two versions of Equation (48):

- `SEIR-TLRF`: using our adaptive growth rates
- `SEIR-tcv`: using traditional time-based cross-validation

For both methods, we import the $r_{t,c}$ estimates generated by the benchmarks into the forecasting procedure, then derive $\beta_{t,c}$ via Equation (50). The results are presented in Table 14 and Figures 16 and 17. The results thus show that TLRF's growth rate estimates can be plugged into a compartmental model and improve its predictive performance. Specifically, our adaptive approach reduces the Median MAE by 12.7% (from 0.395 to 0.345) and the RMSE by 13.6% (from 0.537 to 0.464). These improvements are substantial, especially considering that only about one-third of models evaluated in Cramer et al. (2022) achieved a greater than 10% improvement in relative MAE compared to a simple baseline model.

Method	MAE	RMSE
SEIR-tcv	0.395	0.537
SEIR-TLRF	**0.345**	**0.464**

Table 14 Median MAE and RMSE of SEIR-TLRF v.s. SEIR-tcv

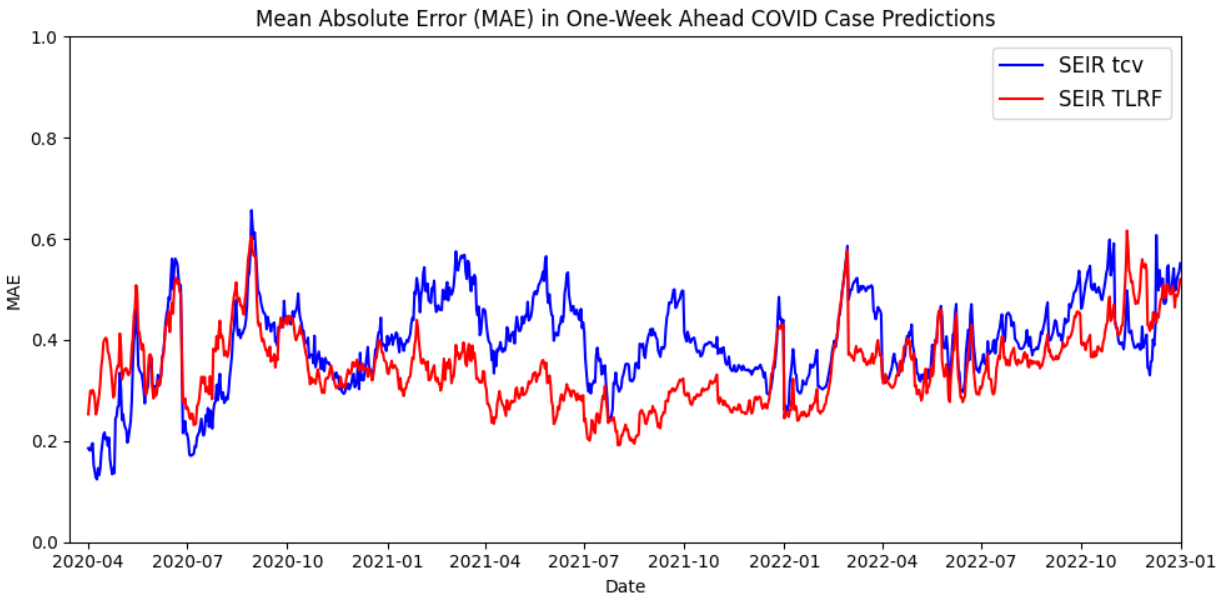

Figure 16 MAE plot of prediction accuracy (SEIR-TLRF v.s. SEIR-tcv)

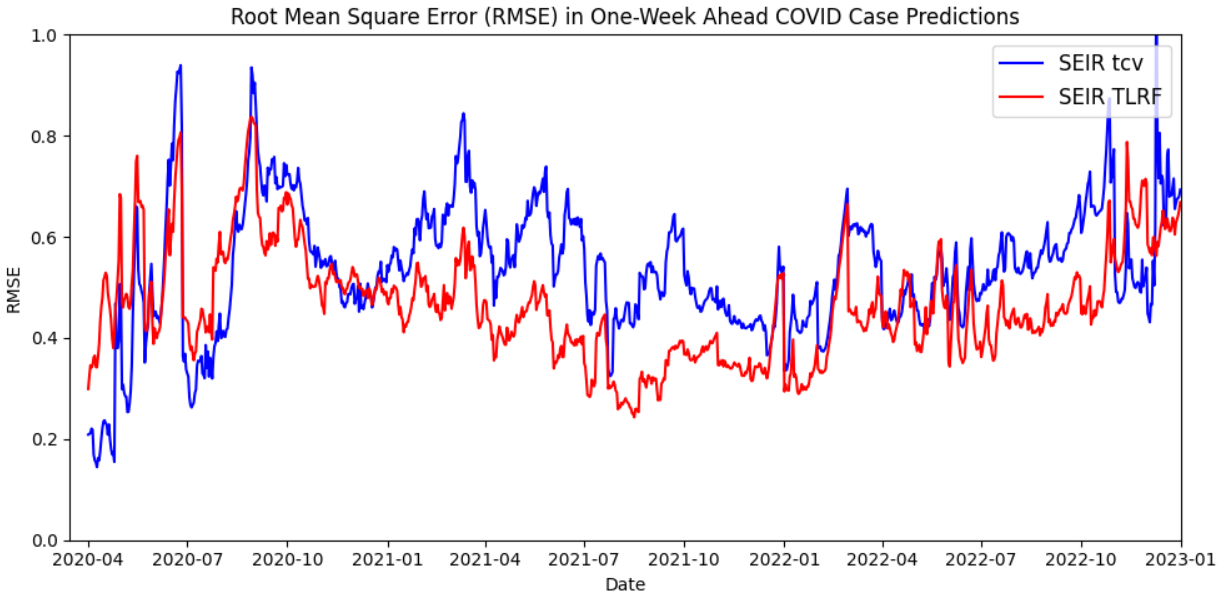

Figure 17 RMSE plot of prediction accuracy (SEIR-TLRF v.s. SEIR-`tcv`)

The parallel implementation of this procedure can be found on our GitHub repository at: https://github.com/wangzilongri/COVID-tracking/tree/TLGRF_major_revision/analysis/coronaSEIR.

Appendix H: COVID-19 Outbreak Detection Tool: Additional Details

The outbreak detection tool reports county-level doubling times on a weekly basis throughout the course of the pandemic. Doubling time, defined as $\frac{\ln(2)}{r_{t,c}}$, converts the instantaneous county-level growth rate $r_{t,c}$ estimated from TLRF to an intuitive metric that is easier to explain to the general public. Specifically, doubling time captures the rate of spread of COVID-19 in each county: a longer doubling time means that COVID-19 is spreading slower, and vice versa. For example, Figure 18 shows the COVID-19 doubling times of each county in the U.S. during the Omicron wave in December 2021 to January 2022. Just before the spread of Omicron variant in early-December 2021, COVID-19 cases were declining, which is evident from the increase in the doubling time from December 05, 2021, to December 26, 2021. However, the Omicron variant started emerging in December end. By early January, the doubling time in most counties was less than 2 weeks. The latest COVID-19 doubling time in each county can be seen at www.covid19sim.org.

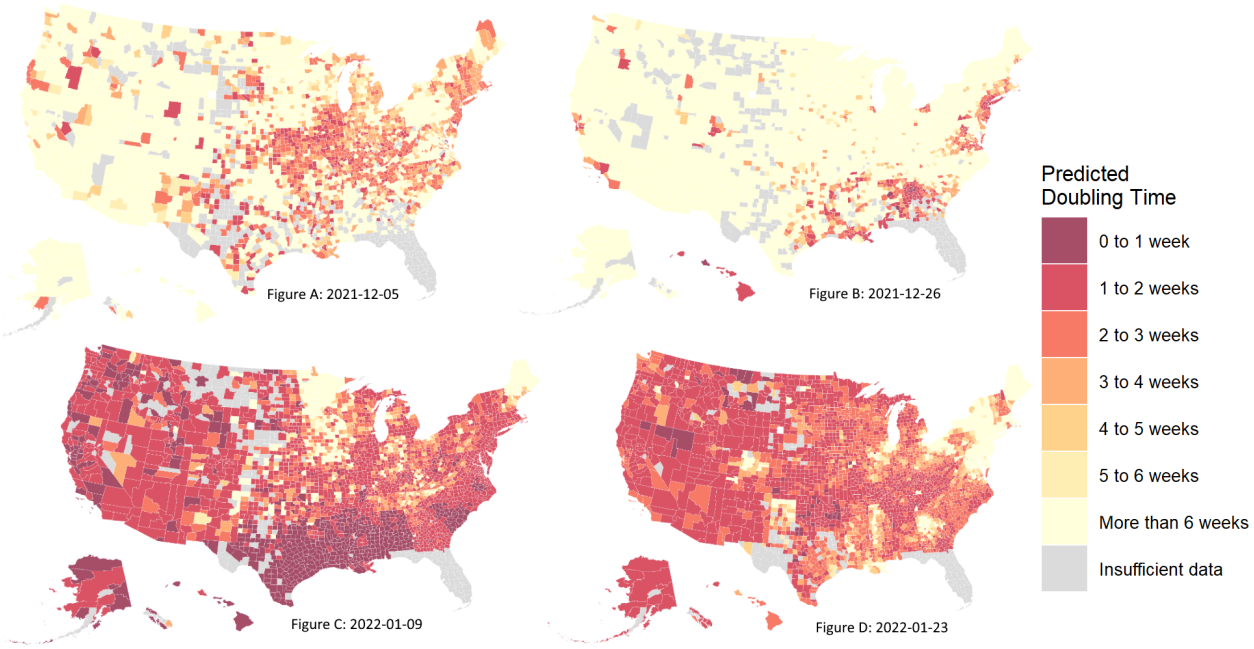

Figure 18 Snapshots of the chloroplast of county-level doubling days on our website at various points in time from 2021-12-05 to 2022-01-23. The doubling day of incident COVID-19 case number at county c on day t is derived from the instantaneous county-level growth rate $\boldsymbol{r_{t,c}}$ as $\frac{ln(2)}{\boldsymbol{r_{t,c}}}$.

Furthermore, we feature a data explorer (see Figure 19): an interactive spreadsheet of doubling days, recorded cases and deaths, vaccination adoption etc. for the more inquisitive members of the public, where they can filter, sort, and search through the data as desired and download it as a csv file. Detailed documentation is publicly accessible on `www.covid19sim.org` and all code and data are open source.

The web platform of this decision support tool, "COVID-19 Outbreak Detection Tool", was released on the 15th of September 2020.

Our outbreak detection tool has been well received since its launch. The website had around 105 daily visitors during the 2020 winter COVID-19 wave, where web traffic peaked at around 2000 visitors a day, and has an accumulated total of 20,000 authentic users from all 50 states. We have also been consulted by public health officials from the states of Maryland, Michigan, and Massachusetts, and helped them use this outbreak detection tool to guide their decision-making during the pandemic. Finally, our outbreak detection tool has also attracted significant public attention and was featured in leading media outlets, including The Voice of America and National Public Radio.

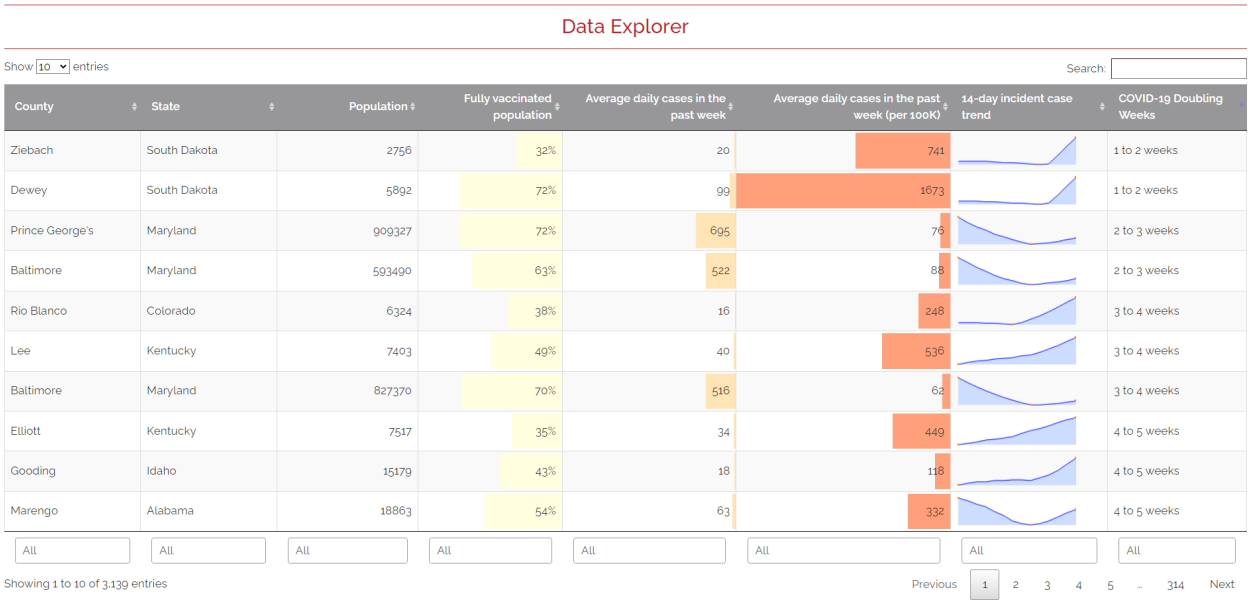

Figure 19 Snapshot of the data explorer on our website, where the user can interactively filter, sort, and search through the relevant county level data and download it as a csv file if desired

References

Arthur D, Vassilvitskii S (2007) k-means++: the advantages of careful seeding. *SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1027–1035 (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics), ISBN 978-0-898716-24-5.

Athey S, Imbens G (2016) Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113(27):7353–7360.

Athey S, Tibshirani J, Wager S, et al. (2019) Generalized random forests. *The Annals of Statistics* 47(2):1148–1178.

Bastani H (2021) Predicting with proxies: Transfer learning in high dimension. *Management Science* 67(5):2964–2984.

Ben Phillips (2020) Coronavirus 10-day forecast. Available at https://covid19forecast.science.unimelb.edu.au/ (Accessed September, 18, 2020).

Bergman A, Sella Y, Agre P, Casadevall A (2020) Oscillations in US COVID-19 incidence and mortality data reflect diagnostic and reporting factors. *Msystems* 5(4):e00544–20.

Centers for Disease Control and Prevention (2018) Centers for Disease Control and Prevention Social Vulnerability Index 2018 Database. Available at https://www.atsdr.cdc.gov/placeandhealth/svi/data_documentation_download.html.

Centers for Disease Control and Prevention (2021a) CDC - variant classifications. https://www.cdc.gov/coronavirus/2019-ncov/variants/variant-classifications.html, accessed 2023-01-01.

Centers for Disease Control and Prevention (2021b) Cdc covid data tracker. Available at https://data.cdc.gov/Laboratory-Surveillance/SARS-CoV-2-Variant-Proportions/jr58-6ysp/ (Accessed February 27, 2023).

Cramer EY, Ray EL, Lopez VK, Bracher J, Brennen A, Castro Rivadeneira AJ, Gerding A, Gneiting T, House KH, Huang Y, et al. (2022) Evaluation of individual and ensemble probabilistic forecasts of covid-19 mortality in the united states. *Proceedings of the National Academy of Sciences* 119(15):e2113561119.

D'Amour A, Ding P, Feller A, Lei L, Sekhon J (2021) Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics* 221(2):644–654.

Hamilton JD (2020) *Time Series Analysis* (Princeton university press).

Julia R (2020) COVID-19 U.S. State Policy (CUSP) Database. URL https://www.evidenceforaction.org/{COVID-19}-us-state-policy-cusp-database.

Kittitas County (2020) COVID-19 Incident Rate Explained. Available at https://nkctribune.com/{COVID-19}-incident-rate-explained-2/ (Accessed: 2021-08-02).

Ma J (2020) Estimating epidemic exponential growth rate and basic reproduction number. *Infectious Disease Modelling* 5:129–141.

McNutt LA, Krug A (2016) Incidence. Available at https://www.britannica.com/science/incidence-epidemiology (Accessed: 2021-08-02).

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

The Atlantic (2018) The COVID Tracking Project. Available at https://covidtracking.com/.

The New York Times (2020) Coronavirus (Covid-19) Data in the United States. Available at https://github.com/nytimes/covid-19-data (Accessed 2020-09-07).

United States Census Bureau (2019) 2019 US Census Gazetteer Files. Available at https://www.census.gov/geographies/reference-files/time-series/geo/gazetteer-files.html.

Zhang LM (2021) Singapore extends stay-home notice to 21 days for travellers from higher-risk countries or regions. *The Straits Times* Available at: https://www.straitstimes.com/singapore/singapore-extends-stay-home-notice-to-21-days-for-travellers-from-higher-risk-places (Accessed November 29, 2023).