

The Brownian Integral Kernel:

A New Kernel for Modeling Integrated Brownian Motions

Béla H. Böhnke¹[0000–0002–3779–3409] bela.boehnke@kit.edu, Edouard Fouché²[0000–0003–0157–7648], and Klemens Böhm¹[0000–0002–1706–1913]

¹ Karlsruhe Institute of Technology (KIT) IPD, Kaiserstraße 12, Karlsruhe, 76131, Baden-Württemberg, Germany

² Siemens Deutschland – Work originated while at KIT, Werner-von-Siemens-Straße 1, München, 80333, Bayern, Germany

Abstract. Brownian motion is widely used to model random processes across various domains. However, many practical scenarios only provide aggregated data over time intervals, rather than direct measurements of the underlying process. This poses significant challenges for accurate modeling, as conventional Brownian kernels are not designed to account for the uncertainty introduced by these aggregates. We introduce the *Brownian integral kernel* (BIK), the first analytical kernel specifically developed to model aggregated data from Brownian motion. Through extensive experiments on synthetic and real-world datasets, we demonstrate the BIK’s superiority in prediction accuracy, uncertainty estimation, and data synthesis compared to existing Kernels. To support adoption, we provide a Python implementation³ compatible with GPy, along with all code and data to reproduce our experiments.

Keywords: Integral Measurements · Learning from Aggregated Data · Integrated Brownian Motion · Gaussian Process Regression · Kernels

1 Introduction

Brownian motion (Figure 1a) is central to modeling various physical and technological processes, such as: (1) The movement of particles in a fluid [7]. (2) The movement and loosening of machine elements due to vibration [13]. (3) The behavior of financial and other markets [17], population behavior and effects. (4) The load profile of electrical grids where producers and consumers with varying loads are plugged in or out of the grid at any time. These processes often involve stochastic uncertainties, which are effectively modeled using Gaussian processes and Brownian kernels [4], enabling synthesis of process data, regression of real-world data with associated uncertainty, and the combination of both, i.e., synthesis of data from partially conditioned models.

However, in many real-world scenarios, data collection pipelines contain “integrators” that implicitly or explicitly aggregate data over time intervals [19]

³ git: <https://github.com/bela127/Brownian-Integral-Kernel>

(Figure 1b). Direct observation of the quantity of interest is not possible in such scenarios. Examples include sensors with inherent integration properties (e.g., temperature sensors with heat capacity) or practical constraints (e.g., smart meters providing 15-minute aggregated load data for value privacy). This aggregation obscures the underlying behavior and increases uncertainty, which conventional Brownian kernels cannot model accurately (Figure 1c). For instance, load forecasting requires precise variance estimation to manage short-term peaks in energy consumption [8, 18, 28]. Without this, providers risk grid instability or legal penalties due to insufficient capacity [3]. Similar problems arise in other domains like disease monitoring and simulation [34]. As emphasized by [5], the importance of integral kernels for modeling integrating processes is undisputed in many fields.

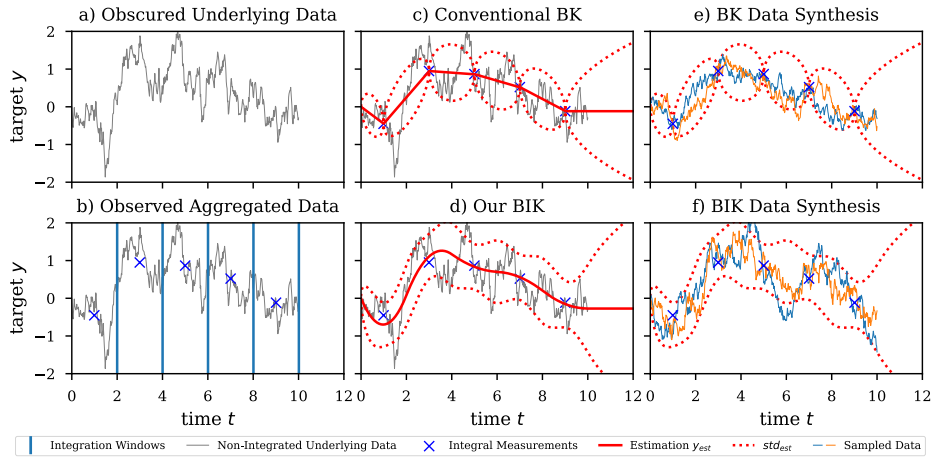


Fig. 1. Comparison of modelling and synthesis of integrated Brownian data with the conventional Brownian Kernel (BK) and our *Brownian Integral Kernel (BIK)*.

While a few integral kernels exist (e.g., RBF Integral Kernel), they are often computationally expensive or lack a direct connection to physical processes, making them unsuitable for integrated Brownian motion.

To address these challenges, we derive the *Brownian Integral Kernel (BIK)*, a novel analytical solution for modeling aggregated data from Brownian motions. The BIK accurately estimates variance and supports Gaussian process regression, enabling better predictions and uncertainty quantification (Figure 1d), as well as data synthesis (Figure 1f). We validate its performance through extensive experiments on synthetic and real-world datasets. Further, to foster accessibility, we provide a Python implementation of the BIK compatible with the widely used [9] framework for Gaussian Process modeling. For brevity of this paper, we provide proofs, theoretical findings, and additional experiments in our extensive Supplementary material.

2 Related Work

Gaussian processes (GPs) are a standard tool for modeling continuous processes, offering flexible kernels that enable accurate regression, uncertainty quantification, and data generation [4]. Their robustness to overfitting and scalability make them well-suited for both small and large datasets [6]. A key strength of GPs is their use of kernel functions, which allow for modeling complex, non-linear relationships without explicit parametric assumptions. The general workflow for using GPs is outlined in Algorithm 1:

Algorithm 1 Generic Use-Case of a Gaussian Processes

- 1: **Input:** Training data $\mathbf{t}_{\text{train}}, \mathbf{y}_{\text{train}}$, Test data \mathbf{t}_{test}
 - 2: **Output:** Predicted mean $\mu(\mathbf{t}_{\text{test}})$, uncertainty $\sigma^2(\mathbf{t}_{\text{test}})$, generated data \mathbf{y}_{gen}
 - 3: **Step 1: Kernel Selection.** Choose a kernel function $k(\cdot, \cdot)$ (e.g., RBF, Matérn, our BIK, etc., or combination).
 - 4: **Step 2: Training.** Train the Gaussian process model on $\mathbf{t}_{\text{train}}, \mathbf{y}_{\text{train}}$ using the kernel $k(\cdot, \cdot)$. Learn hyperparameters θ of the kernel by maximizing the log marginal likelihood.
 - 5: **Step 3: Prediction and Uncertainty Estimation.** For test points \mathbf{t}_{test} , compute the posterior mean $\mu(\mathbf{t}_{\text{test}})$ and variance $\sigma^2(\mathbf{t}_{\text{test}})$ using the trained model.
 - 6: **Step 4: Data Generation from Posterior Distribution.** Sample from the posterior $\mathcal{N}(\mu(\mathbf{t}_{\text{test}}), \sigma^2(\mathbf{t}_{\text{test}}))$ to generate new data \mathbf{y}_{gen} .
-

Despite their strengths, conventional kernels face limitations when modeling aggregated data. The RBF Integral Kernel [5, 29] lacks physical grounding, while numeric integration-based methods [20, 33] and MCMC approximations [34] are computationally expensive. Spectral approximations [10, 30] succeed for smooth kernels but fail for discontinuous ones like the Brownian kernel [22].

The Brownian kernel, in contrast, is closely tied to physical processes such as load profiles and market behavior. However, it assumes direct observations, limiting its utility for aggregated data. Efforts to adapt Kalman filters [12, 15, 24, 25, 31, 32, 35, 36] and integrate numeric methods [16, 20] have resulted in problem-specific or approximate solutions. For more details, see the extended related work in Supp.Mat. 2

Some approaches are out of scope for this work. Time-discrete methods, such as Kalman filters, lack the continuous modeling required for integrated data, while extensions for time-continuous modeling [15, 32] require problem-specific transition functions. Methods like Support Vector Machines (SVMs) do not support uncertainty estimation, and neural networks lack efficient mechanisms for data generation and uncertainty quantification. These limitations make them unsuitable as baselines for this study.

This paper introduces the *Brownian Integral Kernel (BIK)*, an analytical solution for modeling aggregated data. The BIK directly relates to common physical processes and provides exact covariance estimation for integrated Brownian

motion, enabling efficient computation within standard GP frameworks while maintaining a direct connection to physical processes.

3 Problem Statement

For clarity and completeness, the detailed notation and fundamental definitions are provided in Supp.Mat. 1. Although the notation is designed to be intuitive and should become clear from the context, readers are encouraged to refer to Supp.Mat. 1 as needed for additional details.

We assume a data-generating process $B(t)$ that behaves like a Brownian motion. Some real processes that behave like this do not allow observation of their actual value $b(t)$ (realization) at time t . One can only observe average or integral measurements of $B(t)$, while the true value $b(t)$ remains unknown.

Definition 1. *An integral measurement $\mathcal{B}(s, e)$ is the integration over time t of measurements from $b(t)$ from start time s to end time e :*

$$\mathcal{B}(s, e) = \int_s^e b(t) \delta t.$$

Even if $b(t)$ follows a Brownian motion which is per definition erratic, integral measurements $\mathcal{B}(s, e)$ behave differently, this is because integration smoothes the values leading to a smoother function. In consequence the measured value $\mathcal{B}(s, e)$ is most certainly not the true value of the underlying process $b(t)$. However, a Brownian motion model, e.g., a Gaussian process with Brownian kernel $k_{ff'}(t, t') = v_b \cdot \min(t, t')$ assumes that the observed data points are the true values. Fitting such a model on integral measurement $\mathcal{B}(s, e)$ gives the wrong predictive variance of zero at measurement locations. To obtain the correct covariance, i.e., to correctly model the additional variance due to integration, a new kernel is necessary.

4 The Brownian Integral Kernel

We propose to model the additional variance (described in the previous section) directly in a new kernel to solve the problem of mis-estimating the variance. Thus, this calls for a kernel that yields the covariance between two integrated time intervals of a Brownian motion. Following [5] we can derive this kernel by integrating the Brownian kernel:

Definition 2. *The Brownian integral kernel (BIK) is the two-time integration over time intervals (s, e) and (s', e') of the Brownian kernel:*

$$k_{\mathcal{FF}'}((s, e), (s', e')) = v_b \cdot \int_{s'}^{e'} \int_s^e \min(t, t') \delta t \delta t'.$$

This kernel generalizes the Brownian kernel to integral measurements by modeling the higher variance due to integration, which the Brownian kernel neglects. For intuition and proof of how integrating a kernel is equivalent to integrating the underlying process, please refer to Supp.Mat. 3. Further, a kernel needs to fulfill some properties, namely, it needs to be positive-semidefinite [26]. We provide proof for this property in Supp.Mat. 4.

Theorem 1. *The Brownian integral kernel admits the following solution:*

$$k_{\mathcal{FF}'}((s, e), (s', e')) = v_b \cdot \begin{cases} \text{if } s \leq s' < e' \leq e : \\ \quad \mathcal{F}_{tt}((e', t), (s', e')) + \mathcal{F}_{tt'}(s', e') + \mathcal{F}_{t't'}((s, s'), (s', e')) \\ \text{if } s' < e' \leq s < e : \\ \quad \mathcal{F}_{tt}((s, e), (s', e')) \\ \text{if } s' \leq s < e' \leq e : \\ \quad \mathcal{F}_{tt}((s, e), (s', s)) + \mathcal{F}_{tt}((e', e), (s, e')) + \mathcal{F}_{tt'}(s, t') \\ \text{if } s \leq s' < e \leq e' : \\ \quad \mathcal{F}_{t't'}((s, s'), (s', e')) + \mathcal{F}_{t't'}((s', e), (e, e')) + \mathcal{F}_{tt'}(s', e) \\ \text{if } s < e \leq s' < e' : \\ \quad \mathcal{F}_{t't'}((s, e), (s', e')) \\ \text{if } s' \leq s < e \leq e' : \\ \quad \mathcal{F}_{tt}((s, e), (s', s)) + \mathcal{F}_{tt'}(s, e) + \mathcal{F}_{t't'}((s, e), (e, e')) \end{cases}$$

, with $\mathcal{F}_{tt}, \mathcal{F}_{t't'}, \mathcal{F}_{tt'}$ being sub-parts of the primitive function:

$$\mathcal{F}_{tt}((l', u'), (l, u)) = \frac{1}{2}u^2 \cdot u' - \frac{1}{2}u^2 \cdot l' - \frac{1}{2}l^2 \cdot u' + \frac{1}{2}l^2 \cdot l', \quad (1)$$

$$\mathcal{F}_{t't'}((l', u'), (l, u)) = \frac{1}{2}u'^2 \cdot u - \frac{1}{2}l'^2 \cdot u - \frac{1}{2}u'^2 \cdot l + \frac{1}{2}l'^2 \cdot l, \text{ and} \quad (2)$$

$$\mathcal{F}_{tt'}(l, u) = \frac{1}{3}(u - l)^3 + (u - l)^2 \cdot l. \quad (3)$$

We provide a detailed proof of this derivation in Supp.Mat. 8.

The resulting kernel $k_{\mathcal{FF}'}$ calculates the covariance between training data intervals, i.e., between two integrated time intervals. However, during inference, most users are interested in the predictive variance $k_{ff'}^{post}$.

Definition 3. *The predictive variance $k_{ff'}^{post}$ of a Gaussian process is a variance within the original space of the non-integrated Brownian motion. For a given point t' , $k_{ff'}^{post}$ quantifies how likely a prediction $y_{t'}$ is equal to the unobserved ground truth $\hat{y}_{t'}$.*

The predictive variance $k_{ff'}^{post}$ is often used to quantify the uncertainty associated with predictions $y_{t'}$ [27]. To calculate it, we need to calculate the *partly integrated* covariance $k_{\mathcal{FF}'}$.

Definition 4. The partly integrated covariance $k_{\mathcal{F}f'}$ is the covariance between an inference point t' (point in the original space of non-integrated Brownian motion) and an integration time interval (s, e) . It is calculated as a one-time integration of $k_{ff'}$:

$$k_{\mathcal{F}f'}((s, e), t') = v_b \cdot \int_s^e \min(t, t') \delta t.$$

Theorem 2. The partly integrated kernel admits the following solution:

$$k_{\mathcal{F}f'}((s, e), t') = v_b \cdot \begin{cases} \text{if } t' \leq s < e : \\ \mathcal{F}_t(t', s, e) \\ \text{if } s < t' < e : \\ \mathcal{F}_{t'}(s, t') + \mathcal{F}_t(t', t', e) \\ \text{if } s < e \leq t' : \\ \mathcal{F}_{t'}(s, e) \end{cases}$$

with $\mathcal{F}_t, \mathcal{F}_{t'}$ as follows:

$$\mathcal{F}_t(t, l', u') = t \cdot u' - t \cdot l', \quad (4)$$

$$\mathcal{F}_{t'}(l', u') = \frac{1}{2}u'^2 - \frac{1}{2}l'^2. \quad (5)$$

We obtain this partially integrated covariance by one-time integration of $k_{ff'}(t, t')$. The proof follows as a direct result from the proof of Theorem 1, see Supp.Mat. 8. Here, Equations 4 and 5 are an intermediate result (compare with Equations 1 and 2) from Supp.Mat. 8.

We can now calculate the predictive variance $k_{ff'}^{post}$ of the Gaussian process, according to [23] with the standard formula for Gaussian processes. Here we use matrix notation, where K_{ff}, K_{Ff}, K_{FF} is the matrix obtained by using the appropriate kernel $k_{ff'}(*, *), k_{\mathcal{F}f'}(*, *), k_{\mathcal{F}\mathcal{F}}(*, *)$ and K_{ff}^{post} is the result matrix that corresponds to $k_{ff'}^{post}$:

Corollary 1.

$$K_{ff}^{post} = K_{ff} - K_{fF}K_{FF}^{-1}K_{Ff}.$$

5 Experimental Design

We derived the Brownian Integral Kernel (BIK) and highlighted its advantages over existing kernels.

To validate our findings, we compare the BIK in various scenarios against the Brownian Kernel (BK), the RBF Integral Kernel (RBFIK) from [5], and the Brownian Kernel with added white noise (BNK) [2, 14]. The comparisons assess prediction quality, variance estimation, and the plausibility of data generated by a GP conditioned on integral measurements.

5.1 Used Data

For our evaluation, we use data from multiple sources: **Synthetic data:** *Synth* from a Gaussian process with a Brownian Kernel and subsequent numeric integration. **Simulated data:** *Load* generated with a realistic and widely used data generator [21]. **Real-world data:** *Real* provided by a private household, *HIPE* using the High-resolution Industrial Production Energy (HIPE) data set [1], and *Stock*, using daily stock market prices from [11]. More details on the used data can be found in our Supp.Mat. 5. We provide all used datasets which are otherwise not easily accessible, together with the experiment code within our GitHub repository to facilitate reproducibility.

It is important to note that, contrary to the intuition of the name *load profile*, load profiles can have both positive and negative values due to generating nodes such as photovoltaic systems. Especially for short-term modeling of such load profiles, they behave Brownian-like because of randomly moving cloud cover [37]. For long-term modeling, one almost always has additional periodicity (day and night). However, the periods are usually much longer than 15 minutes, which is why they are not relevant for modeling uncertainty due to integration. Our kernel can be used for long-term modeling with standard kernel composition by just adding a periodic kernel. Yet such an extension is not necessary for evaluating the short-term uncertainty effects we are targeting.

5.2 Metrics

We evaluate multiple application-relevant aspects of the kernels. One is often interested in the prediction error. However, in our scenario (where one can only observe integrals of the ground truth but not the ground truth itself) predictions will always be close to the mean within the observed integration interval. Therefore, standard metrics such as Mean Square Error (MSE) are not meaningful when comparing the kernels. Instead, this scenario requires an evaluation that combines prediction and prediction uncertainty. For this, we use the Weighted Mean Absolute Error (WMAE):

Definition 5. *The WMAE quantifies the estimation quality considering the estimated likelihood $l_{GP}(t')$ of an estimation $y_{t'}$ vs the ground truth $\hat{y}_{t'}$:*

$$WMAE = \frac{1}{|\mathbf{T}|} \sum_{t' \in \mathbf{T}} l_{GP}(t') \cdot |y_{t'} - \hat{y}_{t'}|$$

with \mathbf{T} being the set of test points.

The WMAE is an intuitive measure: The estimated likelihood $l_{GP}(t')$ quantifies for a prediction $y_{t'}$ how likely this prediction is. If the model is confident ($l_{GP}(t') \approx 1$) and the prediction is accurate, the WMAE is low. WMAE penalizes prediction errors more when the model is (wrongly) confident, and less when it is (correctly) uncertain. Note that we are interested in the likelihood per data point, thus it is not required to normalize the likelihood across all data points.

While the WMAE is relevant for the prediction accuracy, we can also directly evaluate the estimated uncertainty. For this, we calculate the variance of the ground truth within an integration interval and compare it with the estimated variance, which should be similar. We use the Mean Square Error MSE_{var} between true and estimated variance. Please note, that this metric is only meaningful in combination with the prediction error.

Finally, we evaluate whether the generated data (which is partially conditioned on observed integral measurements) matches the process assumptions: We know that the integral of the ground truth results in the observed integral measurement. Therefore, the integral of generated data should also result in the observed integral. Here, we calculate the Mean Absolute Error MAE_{int} between the integral measurement and the integral value of generated data. We calculate the integral value of generated data by sampling data at the same points in time used to calculate the ground truth integrals and then apply the same windowing procedure, i.e., summing all the resultant values within the integration time interval.

5.3 Experiment Procedure and Model

We evaluate each kernel using the same Gaussian process model. As a model, we use the standard GPy implementation [9]. We also train all configurations in the same way, using the GPy gradient-based maximum likelihood optimizer in its standard configuration [9]. In this configuration, the optimizer performs 5 independent optimization runs, each time with random start kernel parameters. The model then uses the parameters from the best run.

For each dataset, we repeat this procedure 20 times, each time with a different ground truth time series. We then calculate the mean and the variance across the runs of the respective metrics.

6 Evaluation

For brevity, we focus here on the quantitative experiments. However, we feature additional qualitative kernel comparisons and visualizations in Supp.Mat. 6. These are especially useful for domain experts who like to attain an intuitive visual understanding of the superiority of the BIK against other kernels. Further, we provide an ablation study of process and kernel parameters in Supp.Mat. 7, and a theoretical runtime analysis in Supp.Mat. 3.

A comparison of the different approaches, in Table 1, shows that BIK is superior to its competitors in every regard evaluated: The integral of generated data has at least 10 times lower MAE_{int} compared to all baselines. (The RBFIK from [5] cannot generate data.) For our BIK, we provide a more detailed analysis of MAE_{int} in Supp.Mat. 7.

Regarding the prediction with uncertainty, BIK has at least 2 times better WMAE on *Load* and *Synth* $w = 25$ than with all competitors. On *Synth* $w = 50, 100, 200$ the baselines catch up a bit, but BIK still beats them by 30% less

Table 1. Evaluation metrics for different kernels and data sets. Bold entries mark the best result within one metric

| Data | Kernel | Metrics | | |
|-------------------------|--------|----------------------------------|----------------------------------|----------------------------------|
| | | MSE_{var} | MAE_{int} | WMAE |
| <i>Synth</i> $w=25$ | BIK | $5.82 \pm 12.$ | $7.6 \pm .45$ | 3.79 ± 2.9 |
| | RBFIK | $24.9 \pm 25.$ | — | $44.2 \pm 60.$ |
| | BK | $15.6 \pm 20.$ | 72.8 ± 4.6 | 6.9 ± 133 |
| | BNK | $15.6 \pm 20.$ | 73.0 ± 6.0 | 6.9 ± 133 |
| <i>Synth</i> $w=50$ | BIK | $5.87 \pm 14.$ | $3.67 \pm .26$ | 2.33 ± 2.0 |
| | RBFIK | $24.6 \pm 27.$ | — | $13.5 \pm 19.$ |
| | BK | $20.8 \pm 25.$ | 71.1 ± 5.8 | $3.38 \pm 42.$ |
| | BNK | $20.8 \pm 25.$ | 72.4 ± 4.3 | $3.38 \pm 42.$ |
| <i>Synth</i> $w=100$ | BIK | $4.7 \pm 13.$ | $2.0 \pm .14$ | 4.35 ± 3.3 |
| | RBFIK | $20.6 \pm 25.$ | — | $44.8 \pm 62.$ |
| | BK | $17.8 \pm 23.$ | 68.5 ± 5.2 | $5.95 \pm 89.$ |
| | BNK | $17.8 \pm 23.$ | 70.0 ± 4.7 | $5.95 \pm 89.$ |
| <i>Synth</i> $w=200$ | BIK | 5.41 ± 9.2 | $1.12 \pm .09$ | 4.27 ± 3.5 |
| | RBFIK | $22.4 \pm 19.$ | — | $50.9 \pm 71.$ |
| | BK | $20.6 \pm 18.$ | 63.2 ± 5.4 | $5.50 \pm 51.$ |
| | BNK | $20.6 \pm 18.$ | 62.9 ± 4.2 | $5.50 \pm 51.$ |
| <i>Load</i> | BIK | $.50 \pm 1.2$ | $.39 \pm .006$ | $.35 \pm .74$ |
| | RBFIK | $.64 \pm 1.6$ | — | 3.5 ± 7.4 |
| | BK | $.66 \pm 1.6$ | $3.3 \pm .07$ | $12. \pm 26.$ |
| | BNK | $.56 \pm 1.4$ | $3.7 \pm .08$ | $.80 \pm 1.7$ |
| <i>HIPE</i> | BIK | $12.5 \pm 26.$ | $.69 \pm .031$ | 0.42 ± 0.3 |
| | RBFIK | $43.1 \pm 43.$ | — | 8.02 ± 6.5 |
| | BK | $45.5 \pm 44.$ | $5.84 \pm .21$ | $518. \pm 429$ |
| | BNK | $45.5 \pm 44.$ | $5.69 \pm .19$ | $476. \pm 395$ |
| <i>Stock</i> | BIK | 1.13 ± 2.9 | 14.8 ± 1.1 | 0.32 ± 0.3 |
| | RBFIK | 3.65 ± 5.1 | — | 4.48 ± 3.9 |
| | BK | 3.98 ± 5.3 | 63.1 ± 3.2 | $46k \pm 38k$ |
| | BNK | 3.98 ± 5.3 | 60.9 ± 2.6 | $45k \pm 38k$ |

error. Also, the WMAE variance suggests that the baselines are very unstable across data sets.

The MSE_{var} between predicted variance and actual variance during measurement on *Synth*, *HIPE*, *Stock* is 2.6 times better for BIK than any baseline. On *Load* BIK is still 10% better.

Note that the results of BK and BNK are similar on synthetic data. This similarity arises because BNK learns a noise of zero. Indeed, BNK only gives good uncertainty estimates when trained on several different observations of the same data point (or with a smaller time step). With integral measurements, such data cannot be observed, leading to BNK learning incorrect small variances.

One may also notice that the variance in predictive variance (measured with MSE_{var}) is high across the experimental runs for all kernels. This is to be expected, since by chance the ground truth is sometimes simply close to the average within the integration interval, leading to a high deviation. Even with this high deviation, MSE_{var} is still meaningful for comparing the quality of the uncertainty quantification as long as the prediction error is similar. Even though BNK estimates a similar uncertainty as BIK, the metrics MAE_{int} and WMAE show that the uncertainty estimation of BNK is useless because of its much higher prediction error.

7 Conclusions

Integrated Brownian motions are crucial in many physical processes and data collection techniques. The conventional Brownian kernel, while effective for modeling Brownian motion, falls short in capturing the uncertainties associated with integrated data.

This study has tackled modeling integrated Brownian motions precisely, by providing an analytical solution to the novel Brownian integral kernel (BIK). The BIK enables precise estimation of variance associated with the underlying quantity of interest. Further, the BIK is a valuable tool for tasks like regression with uncertainty estimation and for data synthesis partially conditioned on measurements, as shown in our experiments.

Our contributions bridge a significant gap in modeling integrated processes. Our Brownian integral kernel enhances the accuracy and reliability of Gaussian process modeling in such uncertain and dynamic environments by a factor of at least 2 on every dataset and against every baseline. Data synthesis with our integral kernel is better by a factor of at least 10 compared to all baselines.

While our Brownian integral kernel is a substantial advancement, there are avenues for further exploration: First, the challenge of concept drift, i.e., changes of behavior of the underlying ground truth stream, and how to handle it, is a relevant research direction. Additionally, investigating the properties of other integrated processes besides the Brownian integrated process could result in additional new kernels useful for modeling such processes. Finally, using our kernel in applied research could lead to advances in several directions, such as estimating privacy violations and disaggregating load data from smart meters.

Supplementary Material We include detailed notation, additional experiments, and proofs in our Supp.Mat. published on Git Hub⁴.

Acknowledgments. This work was supported by the German Research Foundation (DFG) as part of the Research Training Group GRK 2153: Energy Status Data – Informatics Methods for its Collection, Analysis, and Exploitation and by the Baden-Württemberg Foundation via the Elite Program for Postdoctoral Researchers.

⁴ git: <https://github.com/bela127/Brownian-Integral-Kernel>

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bischof, S., Trittenbach, H., Vollmer, M., Werle, D., Blank, T., Böhm, K.: Hipe: An energy-status-data set from industrial production. In: e-Energy. p. 599–603. ACM (2018)
2. Boyle, P.: Gaussian Processes for Regression and Optimisation. Thesis at WGTN (2007)
3. Cutsem, T., Vournas, C.: Voltage Stability of Electric Power Systems. Springer US (1998)
4. Eric, S., Maarten, S., Andreas, K.: A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions. J. Math. Psychol. (2018)
5. Fariba, Y., T, S.M., Álvarez Mauricio: Multi-task learning for aggregated data using gaussian processes. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2019)
6. Ferris, B., Hähnel, D., Fox, D.: Gaussian processes for signal strength-based location estimation. In: Robotics: Science and Systems (2006)
7. Friz, P.K., Gassiat, P., Lyons, T.: Physical brownian motion in a magnetic field as a rough path. AMS **367**, 7939–7955 (2015)
8. Gilanifar, M., Wang, H., Sriram, L.M.K., Ozguven, E.E., Arghandeh, R.: Multitask bayesian spatiotemporal gaussian processes for short-term load forecasting. IEEE Trans. Ind. Elect. (2020)
9. GPpy: GPpy: A gaussian process framework in python. <http://github.com/SheffieldML/GPy> (since 2012)
10. Jidling, C., Wahlström, N., Wills, A., Schön, T.B.: Linearly constrained gaussian processes. In: NIPS. vol. 30. Curran Associates, Inc. (2017)
11. Kumar, S.: 2019-2024 us stock market data (2024). <https://doi.org/10.34740/KAGGLE/DSV/7553516>
12. Kurtz, V., Lin, H.: Kalman filtering with gaussian processes measurement noise (2019)
13. Lanoiselée, Y., Briand, G., Dauchot, O.: Statistical analysis of random trajectories of vibrated disks: towards a macroscopic realization of brownian motion. Physical Review **98** (2018)
14. Li, Z., Guo, F., Chen, L., Hao, K., Huang, B.: Hybrid kernel approach to gaussian process modeling with colored noises. Comput. Chem. Eng. **143**, 107067 (2020)
15. Liu, F., Gao, Z., Yang, C., Ma, R.: Extended kalman filters for continuous-time nonlinear fractional-order systems involving correlated and uncorrelated process and measurement noises. International Journal of Control, Automation and Systems **18**, 2229–2241 (2020)
16. Longi, K., Rajani, C., Sillanpää, T., Mäkinen, J., Rauhala, T., Salmi, A., Haegström, E., Klami, A.: Sensor placement for spatial gaussian processes with integral observations. In: UAI. vol. 124, pp. 1009–1018. PMLR (2020)
17. Meng, X., Zhang, J., Guo, H.: Quantum brownian motion model for the stock market. Physica A Stat. Mech. Appl. **452**, 281–288 (2016)
18. Mori, H., Ohmi, M.: Probabilistic short-term load forecasting with gaussian processes. In: Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems. IEEE (2005)

19. Musicant, D.R., Christensen, J.M., Olson, J.F.: Supervised learning by training on aggregate outputs. In: ICDM. p. 252–261. IEEE (2007)
20. O’Callaghan, S., Ramos, F.: Continuous occupancy mapping with integral kernels. *AAAI* **25**, 1494–1500 (2011)
21. Pflugradt, N., Stenzel, P., Kotzur, L., Stolten, D.: Loadprofilegenerator: An agent-based behavior simulation for generating residential load profiles. *Journal of Open Source Software* **7**, 3574 (2022)
22. Purisha, Z., Jidling, C., Wahlström, N., Schön, T.B., Särkkä, S.: Probabilistic approach to limited-data computed tomography reconstruction. *Inverse Problems* **35**, 105004 (2019)
23. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. Adaptive computation and machine learning, The MIT Press (2006)
24. Reece, S., Roberts, S.: An introduction to gaussian processes for the kalman filter expert. In: 2010 13th International Conference on Information Fusion. pp. 1–9 (2010)
25. Sarkka, S., Hartikainen, J.: Infinite-dimensional kalman filtering approach to spatio-temporal gaussian process regression. In: Lawrence, N.D., Girolami, M. (eds.) Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 22, pp. 993–1001. PMLR (2012)
26. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT (2002)
27. Settles, B.: Active Learning. Springer Cham (2012)
28. Shepero, M., der Meer, D.v., Munkhammar, J., Widén, J.: Residential probabilistic load forecasting: A method using gaussian process designed for electric load data. *Applied Energy* (2018)
29. Smith, M.T., Alvarez, M.A., Lawrence, N.D.: Gaussian process regression for binned data (2019)
30. Solin, A., Särkkä, S.: Hilbert space methods for reduced-rank gaussian process regression. *Stat. Comput.* **30**, 419–446 (2019)
31. Sun, L., Alkhatib, H., Kargoll, B., Kreinovich, V., Neumann, I.: Ellipsoidal and gaussian kalman filter model for discrete-time nonlinear systems. *Mathematics* **7**(12) (2019)
32. Taghvaei, A., de Wiljes, J., Mehta, P.G., Reich, S.: Kalman Filter and Its Modern Extensions for the Continuous-Time Nonlinear Filtering Problem. *Journal of Dynamic Systems, Measurement, and Control* **140**(3), 030904 (11 2017)
33. Tanaka, Y., Tanaka, T., Iwata, T., Kurashima, T., Okawa, M., Akagi, Y., Toda, H.: Spatially aggregated gaussian processes with multivariate areal outputs. In: NEURIPS. vol. 32. Curran Associates, Inc. (2019)
34. Tanskanen, V., Longi, K., Klami, A.: Non-linearities in gaussian processes with integral observations. In: MLSP (2020)
35. Todescato, M., Carron, A., Carli, R., Pillonetto, G., Schenato, L.: Efficient spatio-temporal gaussian regression via kalman filtering. *Automatica* **118**, 109032 (Aug 2020)
36. Xie, J., Dubljevic, S.: Discrete-time kalman filter design for linear infinite-dimensional systems. *Processes* **7**(7) (2019)
37. Xu, J., Yang, C., Zhao, J., Wu, L.: Fast modeling of realistic clouds. In: CNMT (2009)