Spatial Machine Learning for Lead Service Line Detection
Harvard IACS - BlueConduit

Capstone Project, Fall 2021
Javiera Astudillo, Max Cembalest, Kevin Hare, and Dashiell Young-Saver

Table of Contents

Problem Description

Most cities around the United States have a lurking health hazard spread out across thousands of homes: lead pipes. As a malleable and leak-resistant material, lead was a popular choice for water pipes for thousands of years.[1] Throughout the 20th Century, however, the negative health effects of lead became more apparent (particularly for younger children).[2] To combat these effects, American regulators have sought to remove lead from prominent environmental locations, namely household paint, service lines, and gasoline.[3] In this project, we are concerned with the second of these environments: lead service lines.

If lead could be easily and inexpensively removed, though, this project would have begun and concluded around the time that the U.S. Environmental Protection Agency (EPA) banned lead from new service lines in 1986. Of course, that is not the case. The lurking danger element of lead service lines manifests due to the fact that low levels of lead in water sources may not be immediately obvious. Moreover, the largest challenges occur when lead *leaches* into the water supply from corroded service lines, highlighting that minor environmental changes such as a change in water supply or failing solder can suddenly introduce significantly higher concentrations of lead into a home's water supply.[4] Finally, there are substantial data challenges in this domain. Cities typically have poor records of lead service lines, and many of these service lines are privately owned and without a public record at all. The cost associated with false positives and false negatives are both high, compounding this problem. Because failure to locate lead pipes could lead to lead poisoning for thousands, there is an obvious desire to limit the number of dangerous homes which are never investigated. On the other hand, verifying whether a home has lead pipes requires extensive excavation and incurring a cost of roughly $2,500[5], regardless of the outcome.

In this project, we will be assisting BlueConduit, an Ann Arbor, MI-based startup that develops bespoke machine learning algorithms to assist municipalities and water utilities find lead service lines so that they can be replaced. In particular, this project will focus on ***investigating whether spatial information can be incorporated into an existing machine learning pipeline to improve the "hit rate", or the rate at***

[1] https://www.safeplumbing.org/advocacy/health-safety/lead-in-water.
[2] https://www.cdc.gov/nceh/lead/prevention/health-effects.htm.
[3] https://www.epa.gov/lead/protect-your-family-sources-lead
[4] Service lines are the pipes that connect a residence or commercial property to the common water main. From discussions with BlueConduit, we understand that lead was not a popular choice for larger commercial-use properties due to its relative weaknesses when carrying high-flow water.
[5] https://arxiv.org/pdf/1806.10692.pdf

which digs find lead. To date, BlueConduit's models focus on parcel-level features only, incorporating information such as the year the home was built and other neighborhood-level features to predict whether a particular parcel is likely to have lead. Because these observations are essentially viewed as independent and identically distributed, there is substantial spatial information loss. Intuitively, homes which are near one another are likely to share characteristics which cannot be inferred directly from home-level features. In a literal sense, this is the same intuition behind ML algorithms such as k-Nearest-Neighbors, where observations (or parcels) that are near to one another are assumed to provide information about each other as well.

The outcomes and goals of this project are twofold. First, following discussions with our partner organization, we plan to evaluate multiple approaches to incorporating spatial information to see whether it can improve their classification model. Importantly, from our partner's point-of-view, this investigation can be successful even if spatial information does not improve the model. BlueConduit has rarely tested these spatial methods. Indeed, they have stayed away from direct utilization of spatial features such as longitude and latitude, which are prone to overfitting and failure mode due to the lack of causal structure. During this evaluation phase, though, we would consider a model successful if it improves the "hit rate". As described above, this can be thought of as the precision over the "dangerous" homes. A higher hit rate means more excavations that find lead and fewer which do not, improving the rate at which lead can be removed and reducing the total program costs.

Second, we hope to evaluate whether the spatial information can improve predictions for low data environments. As described below, we will work with data for Flint, MI. Flint has been plagued by a water crisis which began unfolding in 2014 and culminated in widespread distribution of bottled water to residents for years due to the extreme levels of lead within the city. As a result, there are many data points of locations with high levels of lead. As BlueConduit moves from Flint to other cities, they are particularly interested in whether this spatial information can provide lift when few predictors have been captured and many homes have not yet been verified.

Data

A. The Flint Dataset

The primary data for this project have been provided by BlueConduit and focus on Flint, MI. The raw dataset is roughly 55,000 rows and contains 74 features. This proprietary data records a single observation for each parcel (home or commercial property) in Flint. Notably, however, only roughly half of the data have a determination of lead or no lead. These reflect homes which have been verified either before the Flint Water Crisis began in 2014 or were previously known to the city.

There is one notable bias here, which is that there is a real-world selection mechanism going on when each individual parcel is selected and verified. The homes in this dataset may be more likely than the average parcel in Flint to have lead, as those would have likely been targeted for removal. We do not believe that this is necessarily an issue for the external validity of the project for two main reasons. First, our approach is model and feature agnostic. While we will use the Flint data to validate the approach and measure progress, our charge is to assess the viability of the spatial information for Flint and future cities. Second, the overrepresentation of homes with lead may contribute to more of a balanced dataset for our model's purposes. Class imbalance is a well-known and omnipresent issue in machine learning, and if the selection effect were reversed (i.e. the dataset were likely to have very few homes with lead service lines), that would be cause for greater statistical concern.

Over the previous years, BlueConduit and the city of Flint have developed many novel sources of data, including the digitization of over one hundred thousand index cards detailing citywide maintenance.[6] Due to these efforts and the scope of our project, we do not intend to develop additional features apart from the spatial information.

B. Parameterizing distances with OpenStreetMaps

One outside source of data we use is OpenStreetMaps (OSM). A natural challenge of this project is measuring distance. Due to the design of traditional American infrastructure, pipes generally follow streets. So, true Euclidean (or Haversine) distance over a map may not accurately represent the way that spatial information should flow through a city. For example, if two houses share a

[6] https://arxiv.org/pdf/1806.10692.pdf.

backyard but are on different streets, then they might have small Euclidean distance but be connected to two different water mains. Thus, we queried OpenStreetMaps to retrieve walking distances between homes. These walking distances better mimic the "street distance" (Manhattan distance) between homes, since the walking paths follow the streets. Thus, a given house will be more influenced by neighbours in the same street than by a backyard neighbour, which likely has a different water main.

Figure 0: *We use street distances (left) rather than geospatial distances (right) to construct our graphs*

These street distances inform our graph-based models. In the graph-based models, one node is created for each parcel. Then, the nodes are connected with edges that are weighted according to the magnitude of the walking distance between them. Due to the long time it takes to query the Open Street Maps API, we only calculated walking distances between homes that were within 0.5 km (Euclidean distance) of one another. This sparsity ensures that homes that are on opposite ends of the city are not connected by edges and, therefore, not spatially influencing one another in our models. In addition, this type of sparsity can help in regularizing large graph-based models.[7]

C. Exploratory Data Analysis (EDA)

After filtering the dataset for parcels without a determination, there are 26,863 rows, each representing a parcel (home or commercial property) in Flint, MI. Each parcel has 74 described features, such as the market value, size, location (latitude/longitude), year built, voting precinct, etc. Because our dataset size is relatively small, we are currently not too concerned about efficiency and computational issues.

[7] https://arxiv.org/pdf/1706.02216.pdf

pid int64	Property Zip Code float64	Owner Type object	Owner State object	Homestead object	Homestead Percent float64	HomeSEV
4012482018	48503	Private	MI	Yes	100	18400
4013226009	48503	Private	MI	Yes	100	11800
4012476011	48503	Private	FL	No	0	0
4012481022	48503	Private	MI	Yes	50	4550
4013226025	48503	Private	MI	Yes	100	12800

Table 1: *The head of our dataset, which shows the one-row-per-parcel structure of the data*

In addition to our features, we have a column for our target: a binary indicator of whether the home has dangerous pipes (1) or safe pipes (0). In the full dataset, which includes the homes that have been investigated in Flint, we found that 10,266 parcels had dangerous pipes, or about 38% of all homes. Because this is a relatively large proportion of all homes, we don't have to treat the target as a "rare" outcome in our models. Additionally, we understand that a full inventory of parcels in Flint would capture significantly more parcels which are either (a) abandoned or (b) are very unlikely to have lead due their date of construction and have therefore not been investigated.

Commercial Condition 2013	26760
B_aggregate_income	4666
B_imputed_value	1500
B_imputed_rent	1500
B_hispanic_household	1500
Housing Condition 2012	400
Residential Building Style	114
Housing Condition 2014	109
USPS Vacancy	59
Owner State	26
Longitude	6
Latitude	6
Use Type	2
Zoning	2

Table 2: *Missing value counts*

Table 2 shows the missing value counts for different features in our data. The highest count is for "Commercial Condition." However, the vast majority of these parcels in the dataset are residential, and this feature is coded as "missing" for residences. So, it's not truly missing but, rather, not applicable to residences. The other features with large missingness rates are the "B_" features. These are all features that were imported from the American Community Survey (part of the US Census). The missingness here is likely from homes that could not be linked to ID's from the American Community Survey. If these features are found to be important, we may go back to the census to see if we can successfully match these homes to American Community Survey data and fill in the missing values with the true values.

Papers published by BlueConduit show that “year built” is consistently the most important predictor of lead in their models. So, we explore this variable with particular interest. A histogram of “year built” is shown in Figure 1.

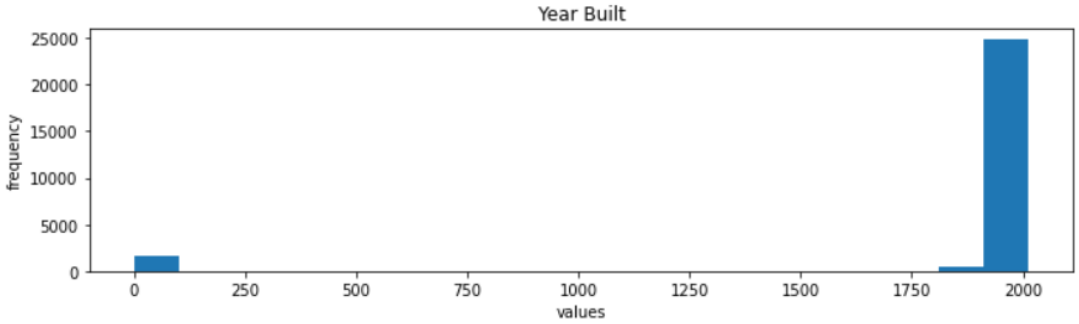


Figure 1: *Histogram of “year built,” including the nonsensical values present in the data*

We saw that there were a fairly high number of nonsensically small values - homes that were built before the year 100 A.D. In total, there are 1,629 homes with nonsensically small year built values. We asked the partner why these values are present, and they reported that some city records are simply unreliable and that they treat these values as “missing.” The Baseline XGBoost model is able to take this missingness in stride by splitting into different cases whether the year built feature is present or absent.

When looking just at the sensical year built values, we find an interesting pattern, visualized in Figure 2.

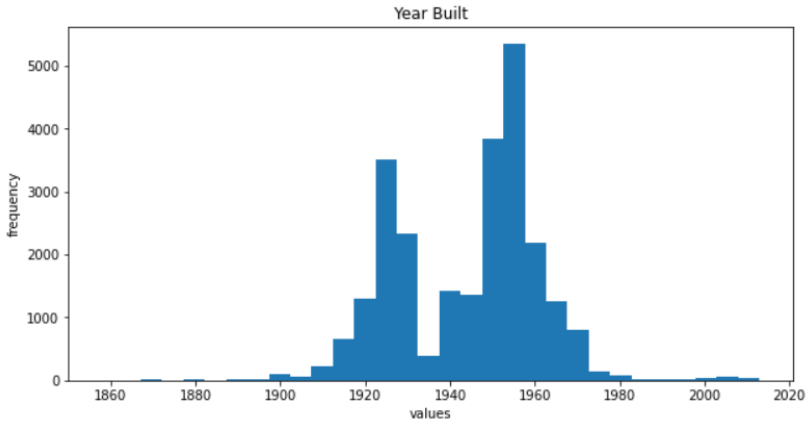


Figure 2: *Histogram of “year built,” only including the sensical values*

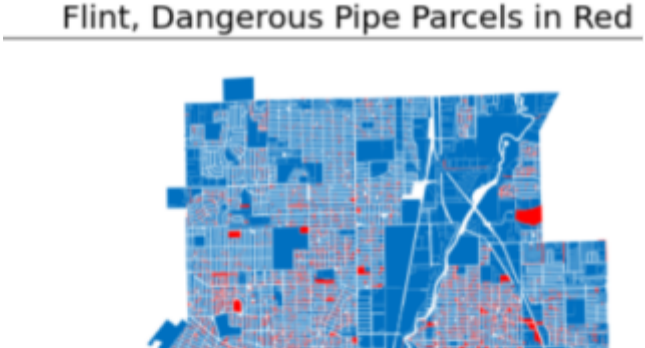


Figure 2 shows that year built appears to be somewhat bimodal, with large buildup of homes prior

to 1935 and immediately following 1945. This is likely due to historical circumstances, such as a lower number of homes built during the Great Depression (1930s) and with a boom after the Second World War (1950s). It may end up being the case that these two eras (pre Depression, post WW2) are fairly predictive of dangerous pipes.

To test this theory, we found the rate of dangerous pipes within both eras. Among pre-Depression homes, 88.5% had dangerous pipes. Among post-World War 2 homes, only 7.2% had dangerous pipes. So, it's clear that year built is highly predictive of lead presence. This explains why it's such an important feature in BlueConduit's current models.

Finally, in Figure 3, we visualized parcels where there is known lead in Flint (lead parcels shown in red). We see that lead pipes are distributed throughout the city; however, there is some variation in lead pipe density by neighborhood. This suggests that using geospatial information could help improve the BlueConduit models. In future map visualizations, we will vary color intensity by predicted probability of lead, and we will refrain from coloring the large commercial parcels (to preserve the granularity of seeing the smaller residential parcels).

Literature review

A. Previous work by BlueConduit

The very first work of BlueConduit with the Flint Council dates back to 2016 and is reviewed in Chojnacki at el. (2017)[8]. In this work, they frame Flint's water lead contamination, examine the water sampling process in detail and propose an initial model for house prediction lead presence. Their work also includes a discussion on how selection bias is a concern. Their models directly impact public health and, thereby, must address interpretability and accountability.

Their initial proposed model predicts a lead presence probability based on parcel dataset, service line dataset, and census dataset, for a total of 71 features. Their target variable uses water testing datasets based on residential and sentinel water testing programs, gathering information for around 15,000 parcels. Their findings show that the most predictive features for predicting lead levels include home

[8] https://arxiv.org/pdf/1707.01591.pdf

value, demographic data from the census bureau and property age.

Their next work (Abernethy et al. 2018[9]) describes their model results in the context where the City of Flint took action in the water crisis. In 2016, Flint's Mayor contracted the Flint Fast Action and Sustainability (FAST Start) team. Their task was to remove as many hazardous service lines as possible up to the funding level. This publication analyses how their model considerably surpasses the FAST Start strategy and empirically shows the resources saved through their strategy.

They update their classification framework to handle unobserved variables with a Bayesian spatial model. Further, their targets are built based on excavations in conjunction with water samples lead level (ppb) in this new setting. As of September of 2017, the FAST Start had carried out 6,506 home excavations. Consequently, they pose their strategy in an active learning framework, simulating the actual scenario process. Every time they make a new batch of discovery excavations, they update their dataset and their predictions accordingly.

They quantify the cost savings between their strategy and Flint FAST Start's actual home selection during 2016-17. In simulating their results over Flint, MI, BlueConduit found that their strategy reduced the rate of costly unnecessary replacement visits from 18.8% (actual) to 2.0% (proposed). Suppose 18,000 total planned service line replacements; this would translate into savings amounting to as much as $11M from current spending. This is approximately equivalent to 2,100 additional homes in the city that would receive safe water lines.

B. Additional work in spatial machine learning

- Bayesian Spatio-Temporal Gaussian Process [Gilanifar et al. (2019)]

  We found this paper when researching spatial models that take time into account as a feature. The temporal dimension is a very relevant consideration for our project since our EDA and the trained XGBoost model both show the importance of YEAR BUILT as a feature.

- A Comprehensive Survey on Graph Neural Networks [Wu et al. (2019)]

  This resource has helped us compare different graph neural network architectures and problem-solving paradigms at both a high-level

[9] https://arxiv.org/pdf/1806.10692.pdf

and on the level of technical details.

- GraphSAGE: Inductive Representation learning on Graphs [Hamilton et a. (2018)]

  This has been the most successful Graph Neural Network architecture used so far. We hypothesize the strength of GraphSAGE in our setting relies on its sub-sampling of neighbors to improve runtime, and its ability to train via batches while the other graph methods train on the whole graph each train step.

Evaluation Framework & Cross-Validation

Because we have considered a variety of models and frameworks, a consistent evaluation framework is of the utmost importance for this project. The evaluation framework for this project relies on two key concepts: spatial cross-validation and the hit rate curve.

First, because of the spatially-connected nature of the data, we aggregate data for cross-validation and evaluation by geographic partitions. This both prevents target leakage and simulates a more realistic environment for investigating parcels, where cities are more likely to select entire blocks or neighborhoods for investigation rather than working parcel-by-parcel in a scattershot manner. Additionally, we employ a custom loss metric, the hit rate curve, which is connected to the precision over the class of homes with lead pipes, but allows for a more dynamic understanding of the potential thresholds for parcel investigation, once again simulating a more realistic choice for a policymaker. Both of these design decisions and implementations are described below in greater detail.

A. Spatial Cross-Validation

When designing our cross-validation strategy, it's essential to mind two key components that will bias our results if not adequately incorporated in our evaluation process: the spatial nature of the problem and the particulars of pipe excavation.

The spatial component regards the dependence structure in the data.[10] It presents itself as the high correlation of lead pipe presence between neighbour houses in the current set. Observation-level training-validation splits rely on the assumption

[10] David Roberts, et al., "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," *Ecography*, Vol. 40, No. 8:: 2016. Available at https://onlinelibrary.wiley.com/doi/10.1111/ecog.02881

of independence between observations in our dataset. However, we know that our observations (each home) are not independent. Indeed, there are substantial spatial dependency structures (nearby homes are similar). Therefore, if we naively split train and test samples with simple random sampling, we could yield overly optimistic prediction error estimates. This is what we call the "target leakage" effect. Models appear more accurate than they are, enticing us to have more faith in their predictions than is actually warranted.

The second component concerns designing train-test splits that are consistent with the pipe excavation scenario. In reality, pipe digging will be carried at "blocks" or "neighbourhood" levels. This means that the city council targets locations with many positively predicted lead pipes houses and digs all those candidates at once. Doing so makes efficient use of excavation process resources. Therefore, we impose a spatial structure on the train and test sets during evaluation, uncovering labels by “block” or geographic area. One consequence of this methodology is that a home with a predicted probability of e.g. 0.9 may be investigated earlier than one with a predicted probability of 0.95 if the latter is in a less lead-dense neighborhood. While this decreases performance, it brings evaluation of the baseline and our methods closer to reality.

A common solution addressing both these concerns of hold-out independence and accurate representation of splits is spatial cross-validation. It achieves unbiased error and better parameter estimates[11] by splitting the data into "blocks" by location. In the current scenario, we apply this methodology by dividing the city of Flint into spatial hexagons and assign each of them either to train or validation split. That way, we reduce the correlation between train and test samples, as most neighbours will be located either entirely within the train or entirely within the test set. Further, the digging will be executed at the hexagon level, emulating the actual scenario. We repeat this procedure multiple times (folds) to perform inference over various samples of the hold-out error metric. Furthermore, we try different train and test proportions to represent the different phases of lead pipe excavation in a city, starting with very few labels. Then, we progressively uncover labels to simulate pipe excavation.

The resulting cross-validation strategy is depicted in the following plots:

[11] Roberts et al. (2016). Available at https://onlinelibrary.wiley.com/doi/10.1111/ecog.02881.

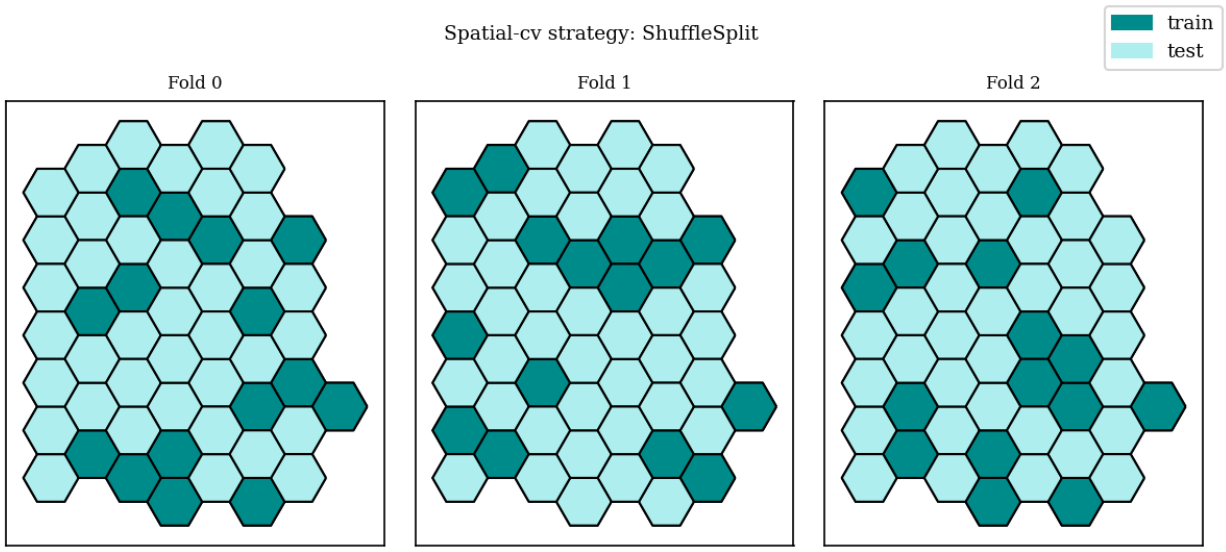


Figure 4. Spatial cross-validation train-test split [folds=3, hex_resolution=22, train_size=0.3]

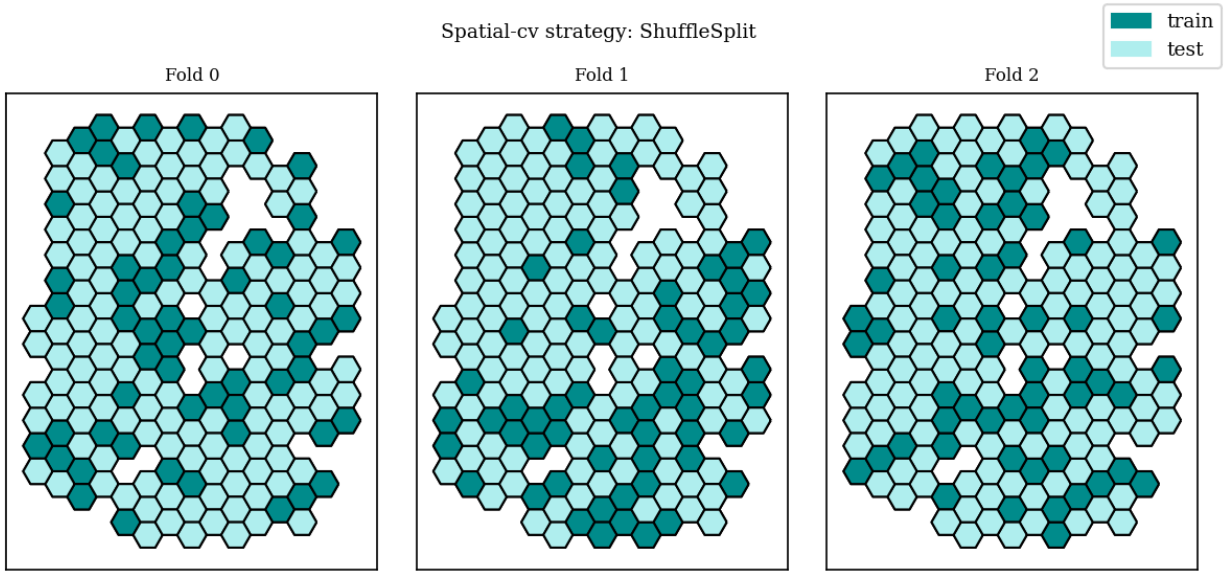


Figure 5. Spatial cross-validation train-test split [folds=3, hex_resolution=47, train_size=0.3]

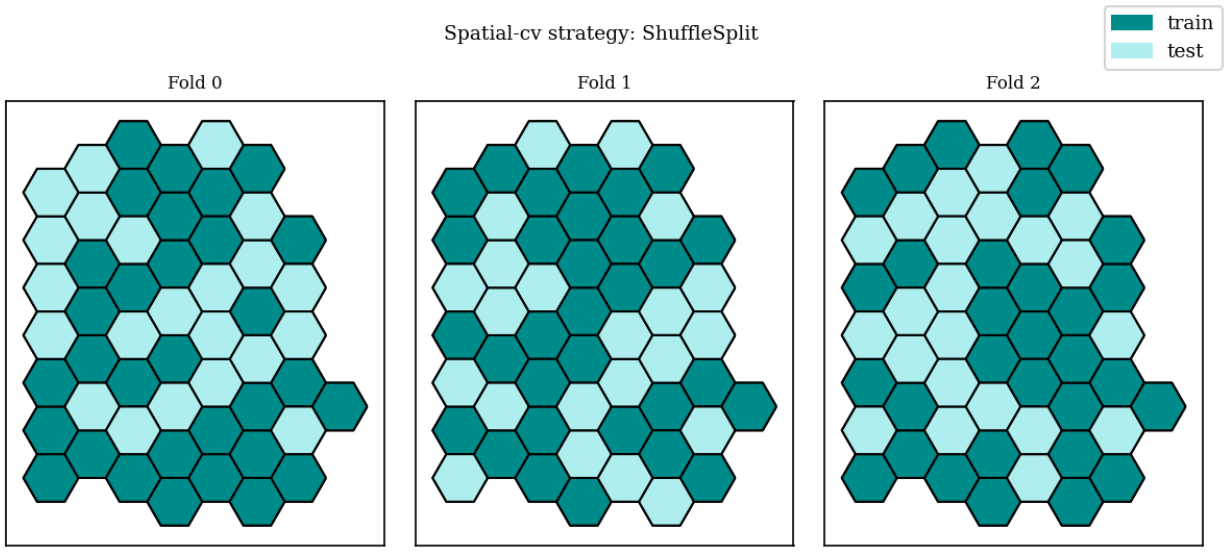


Figure 6. Spatial cross-validation train-test split [folds=3, hex_resolution=22, train_size=0.6]

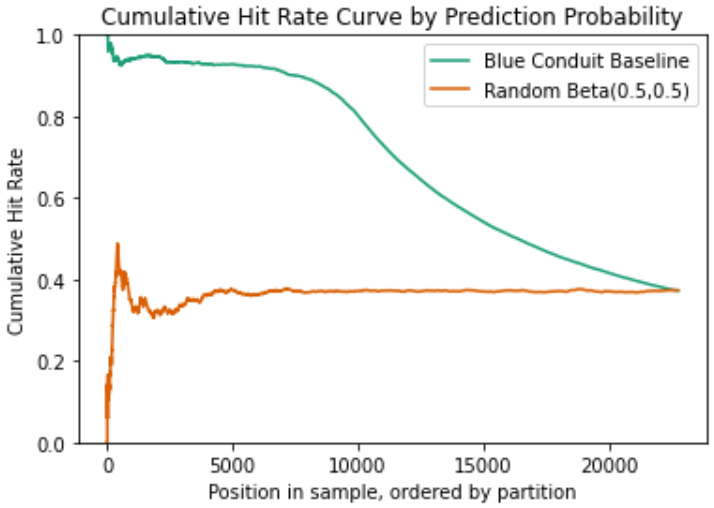
B. Evaluation via the Hit Rate Curve

To properly evaluate our models, we will use a BlueConduit customized loss metric referred to as the "hit rate curve". First, it is important to define the hit rate. As described in Abernathy et al. (2018), the hit rate is the, "percentage of homes visited for replacement that required replacement".[12] In technical terms, this is the precision over the class of homes having lead. Under the traditional machine learning paradigm, the hit rate would be calculated across the entire test set, or across multiple test sets and then averaged during cross-validation. However, doing so would require setting a threshold for investigation. Municipal officials likely have a variety of preferences and care about how the model would do under different conditions. For example, if setting the threshold at 0.5 would be expected to uncover only half of the lead, that would likely be viewed as less-than-optimal compared to a threshold of 0.4 that uncovered 90% of the lead, even if the hit rate were higher in the first instance.

Rather than evaluate models via a single number, BlueConduit makes use of a richer representation of the outputs: the prediction probabilities. In this framework, the "hit rate curve" essentially demonstrates the hit rate for various prediction probability thresholds. Naturally, most trained models yield a high hit rate when

[12] Abernathy, Jacob, Alex Chojnacki, Arya Farahi, Eric Schwartz, and Jared Webb, "ActiveRemediation: The Search for Lead Pipes in Flint, Michigan," *KDD '18*, London, United Kingdom, 2018. Available at https://arxiv.org/pdf/1806.10692.pdf.

the threshold is close to one and converge to the test set average proportion of the positive label as the threshold approaches zero. One implementation detail is that rather than plot the hit rate vs. the prediction threshold, we have elected to plot the hit rate vs. the parcel visited, following conversations with our partner. Thus, if 20% of the test set has a predicted probability of lead of greater than 0.9, this will still be represented at roughly the 20th percentile of the x-axis.



Here, we demonstrate a visualization of the Hit Rate Curve for the Blue Conduit Baseline model (described below) as well as a non-model which guesses according to a Beta(0.5, 0.5) distribution to both demonstrate the visual as well as the performance of the baseline relative to random guessing.[13]

Figure 7: *Hit rate curve for Baseline model*

Appendix A demonstrates a side-by-side comparison of the Hit Rate Curve organized by the prediction probability and parcel investigated for the BlueConduit baseline model.

One weakness of the standard Hit Rate Curve described above is that it does not take into account the reality of investigating lead throughout a city. In general, cities would prefer not to receive a list of 100 parcels to dig on 100 separate city blocks, even if the model outputs those parcels as the highest probability of lead. There are economies of scale to removal crews, city shutdowns of traffic, and pooling of municipal resources. Thus, we alter the algorithm to investigate by cross-validation partition, which works as follows:

[13] A Beta(0.5, 0.5) is used for comparison as it tends to push the distribution towards 0 and 1, similarly to a confident XGBoost model. Subsequent testing, however, suggests that selection of distribution parameters does not measurably alter the results.

Algorithm 1: *Hit Rate Evaluation by Spatial Partition*

By-Partition Hit Rate Curve Algorithm

- Initialize threshold (e.g. 0.9)
- Repeat until all parcels investigated:
  - Calculate count of parcels in partition with prediction probability exceeding threshold.
  - Order by partition.
  - For each partition:
    - Investigate all parcels above threshold; remove from partition.
  - Increment threshold (i.e. threshold = threshold - 0.1)

This methodology is less efficient than the original due to two underlying mechanisms. First, some parcels which have a very high probability of lead may not be in large or high-lead areas. In the initial algorithm, those homes will still be equally prioritized. In our more realistic evaluation scenario, though, these homes will be somewhat de-prioritized by being put into a partition that is investigated in a later time frame.

Second, our framework relies on the total number of expected hits in a partition. Partitions have equal geographic size but not equal populations, and thus partitions where there may be greater geographic economies of scale will be preferred. We also believe that this is a reasonable and realistic assumption because a dense, high lead area is likely to be identified and prioritized before a single parcel in a non-lead dense area. Below, we see that the new evaluation algorithm causes the Hit Rate to decrease in the more realistic scenario, highlighting the tradeoff between following the model entirely and balancing the costs of shutting down infrastructure such as water or roads for some amount of time.

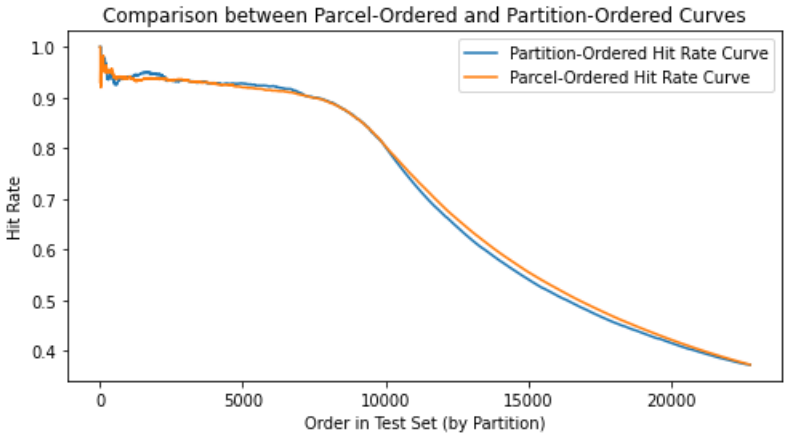


Figure 8: *Hit rate curve comparison of algorithms*

C. Calculating cost savings and days of cumulative lead exposure

While the hit rate satisfies a technical need to align and compare models, it is at best a proxy for cities' and BlueConduit's true goal and business proposition: reducing the cost to cities per lead pipe removed (thus allowing a city on a fixed budget to remove more lead for the same fixed cost) and decreasing the cumulative number of residents exposed to lead on any given day. Fortunately, both of these metrics correspond to a higher hit rate. By identifying the neighborhoods and parcels with the highest likelihood of lead, the city will ideally only perform successful digs[14], extending their funding to the maximum possible limit. Second, every unsuccessful dig for lead requires time for a construction crew to process the parcel, meanwhile another parcel which truly does have lead can still poison residents for another day. We show these results by focusing on the cumulative costs at various intervals of "lead digs". That is, we estimate the total costs for each of the modeling approaches below to remove a certain amount of lead. One practical challenge in this approach is that our dataset is known, and thus we can more easily calculate recall. Testing in these environments is important to understanding the dynamics of lead discovery, but may not reflect the actual, live performance of these algorithms.

Modeling

A. Baseline Model

Our chosen baseline model consists of BlueConduit's current model, mentioned in the previous literature review. We agreed on this with our partner since this project aims to improve based on what they have built so far by incorporating spatial modelling.

More specifically, the baseline model consists of an XGBoost, with 102 input features, spanning parcel, service line and census data. Their target is a binary variable indicating whether there's a lead pipe presence or not. We are mainly concerned about this model's hit rate, representing the lead rate discovery behaviour of the samples ordered by their predicted probability of having a lead pipe. We will consistently compare our developed models with this same metric. In

[14] From Blue Conduit and the municipality's perspective, a dig which finds lead is "successful" as it means that existing lead is removed from active service.

Figure 9, we depict the hit rate of BlueConduit's baseline model alongside some basic comparison models included as a reference.

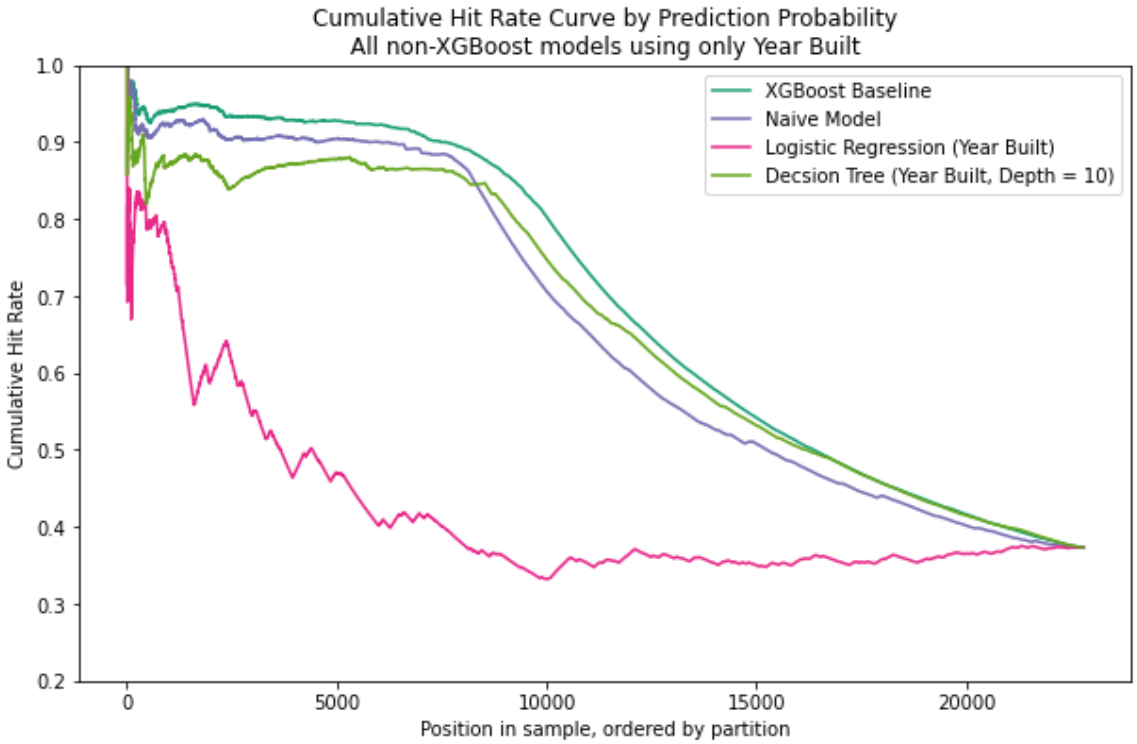
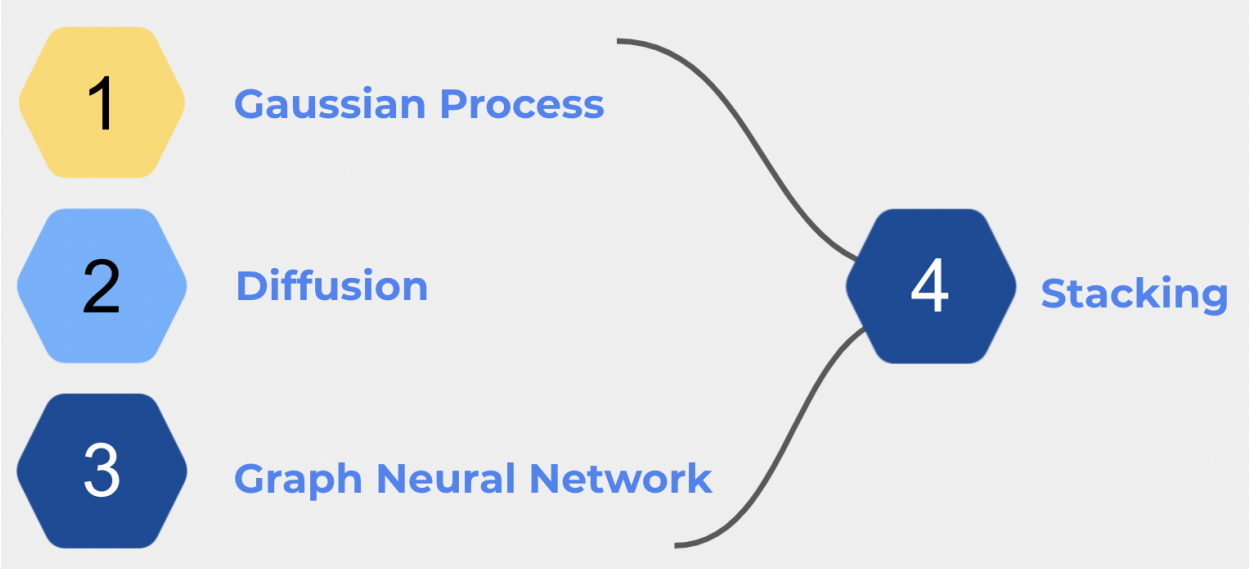


Figure 9. *Baseline model and other additional model references.*

In Figure 9, the BlueConduit XGBoost baseline model is represented by 'Jared Baseline'. Based on our EDA findings, we included reference models trained only on the year predictor ("Logistic Regression w/Year" and "Naive Model w/Year") to get a sense of how much information is conveyed by the year. Also, we included a simple decision tree as a comparative reference of the XGBoost baseline model.

It's clear that each of our naive models have lower hit rate curves than the baseline BlueConduit XGBoost model. Our goal will be to use spatial information models that can raise the hit rate curve beyond BlueConduit's current XGBoost model.

B. Overview of Meta-Model



Our group has working implementations of three different classes of spatial models so far: 1) a Gaussian Process (section C), 2) a Diffusion algorithm (section D), and 3) a few different Graph Neural Network architectures (section E).

Each uses a different feature set, and each has useful properties. For example, the Gaussian Process uses *only* latitude and longitude, and is therefore useful early in the data-collection process as a baseline spatial model. The Diffusion algorithm consists of only slightly tweaking the XGBoost baseline estimates with a weighted average of neighboring probabilities, resulting in a smoothing effect that seems to yield our strongest results so far. The Graph Neural Network leverages *all* features collected so far, and learns* how to estimate a probability of lead for a home based on all available information about the nearby homes.

We intend for our final model to be an implementation of Stacking (section F) which will make a final prediction of lead for each home based on the probabilities of lead from XGBoost as well as the probabilities of lead from our 3 spatial models. In Section F, we describe (only hypothetically, we haven't implemented this so far) what a decision tree might look like as a stacking model for our data.

*to what extent this information is robustly and faithfully "learned" depends on the Graph Neural Network architecture implemented. We have reason to believe GraphSAGE is the most straightforward, reliable choice. The other deep architectures are somewhat promising but likely unreliably overfitting to the data.

C. Gaussian Process (GP)

For a first pass at a spatial model, we chose to use a Gaussian Process to predict the probability a parcel contains dangerous materials, given only latitude and longitude as features.

From our EDA we could see the homes with lead are generally clustered in the middle of the city. Our GP would need to be able to identify the spread of these regions and construct a set of highly nonlinear decision boundaries to account for the twists and turns of the structure of the underlying streets and neighborhoods.

The biggest drawback of a GP we have encountered is its runtime, which is $O(n^3)$; therefore we have not trained a GP on all n=26k homes. Instead, we trained 100 different GPs on a set of bootstrapped data samples, with n=2,000 homes in each sample. Figure 5a shows to what extent the fit of the GPs varies when changing the particular sample of homes they were trained on, and Figure 10b shows what the average of the 100 GP predictions looks like.

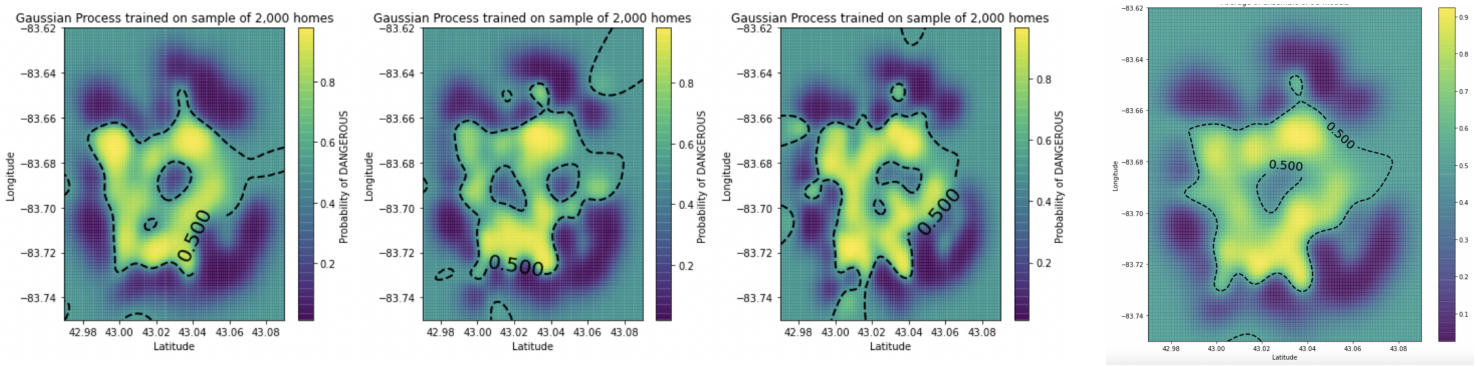


Figure 10a *(left three): Three of the 100 trained GPs, plotting the predicted probabilities of lead across the latitude/longitude values spanning Flint, MI. Each of the GPs identifies that lead in Flint is mostly concentrated towards the center of the city.*

Figure 10b *(rightmost image): Predicted probabilities of the AVERAGE of the 100 trained GPs. This average of GPs had an AUC of 0.8288, higher than the vast majority of individual GPs in the 100.*

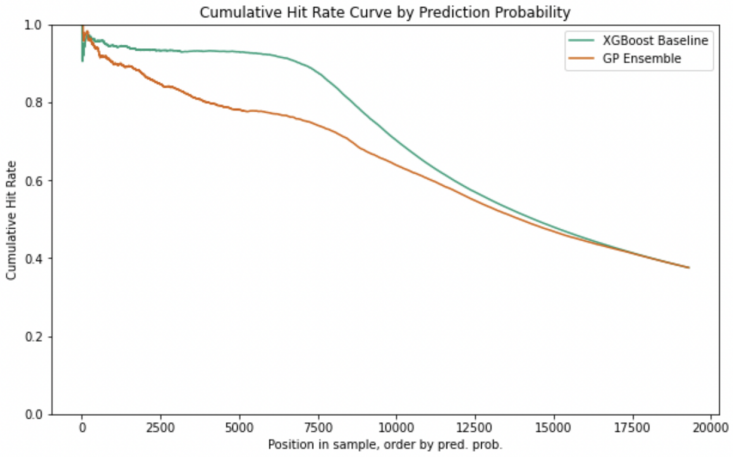


Figure 11: *Hit Rate Curve for the Gaussian Process Ensemble (with samples ordered the old way, by individual predicted probability of lead. Our newer hit rate curves on our graph models are calculated via the Partition-Ordered hit rate curve, described in section V.B)*

Milestone 3 updates: We decided to extend our gaussian process with a third axis: time!

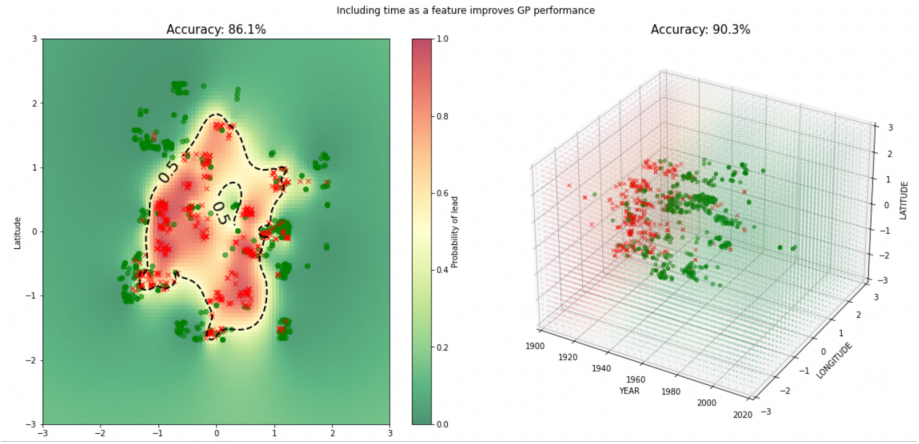


Figure 12: *comparing the predictive accuracy on a test set when Year Built is included as a feature (right), compared with our original gaussian process using only latitude/longitude (left).*

In addition, we decided against using the ensemble-of-100-bootstrapped-gaussian-processes approach; in order to evaluate fairly across models using the same train/test splits for each model, we wanted to restrict the amount of data our GP had access to down from the entire dataset it used originally to the same level provided to our other models, which made the proposition of averaging across 100 GPs unfeasible now that it had less data.

For our stacking model, we provided both the predicted probabilities of lead (left) as well as the *variance* of the model predictions (right) for each experiment we ran. The intention here was for the variance to provide some signal of uncertainty that might assist our stacking model in choosing which spatial model to draw from and when. The variance indicates both aleatoric and epistemic uncertainty - we have high variance both when we have seen little data from a region as well as when we have seen lots of conflicting data (i.e. many homes with lead, many homes without lead) in a region.

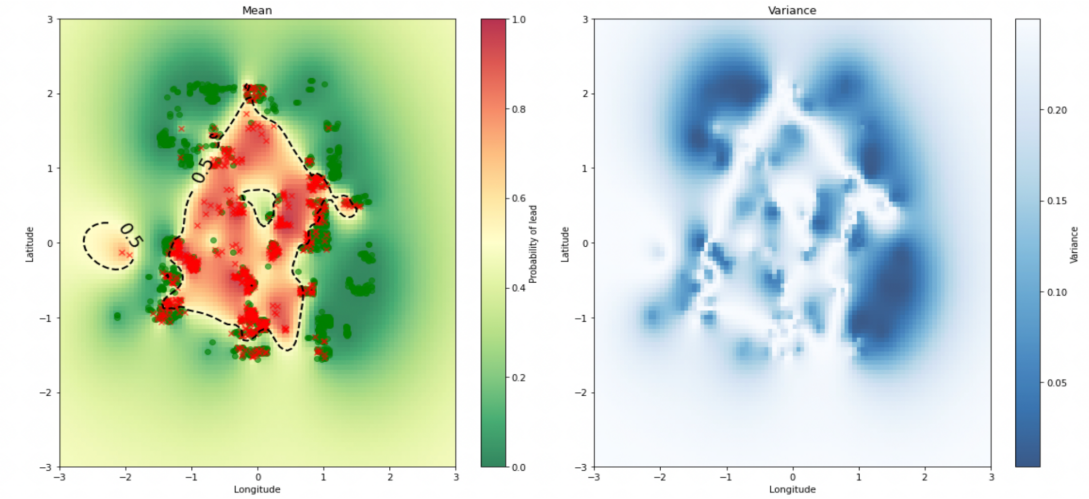


Figure 13: *Visualization of ensemble means and variances from spatial GP*

D. Diffusion

A second attempt to incorporate the spatial information into the modeling process has featured a "diffusion" process. Inspired by anisotropic diffusion to blur images and opinion dynamics over networks, the intuition for this approach is that neighboring homes should share information about one another's lead status.[15] Cities do not grow as random entities, with homes inserted at random times and places. Instead, blocks and neighborhoods tend to be built in similar eras, and thus with similar building materials. Thus, if the baseline parcel-level features-only model outputs a low predicted probability of lead for a single parcel on a block with many high-lead predictions, either the parcel-level features provide some indication of a different construction material or time frame or the model may be incorrectly focusing on a non-relevant feature in the data. In the ideal setting of the diffusion model, the process will "blur" the prediction probability to make it more like its neighbors, hopefully identifying additional high-risk parcels.

[15] https://en.wikipedia.org/wiki/Anisotropic_diffusion; Phillip Converse, "Information FLow and the Stability of Partisan Attitudes," *Public Opinion Quarterly*, Vol. 26, No. 4, 1962, available at: https://academic.oup.com/poq/article-abstract/26/4/578/1868671.

We have begun to investigate two possible formulations of this diffusion process, both of which are graph-based. As described above, nodes are individual parcels, and edges are formed on homes which are determined to be neighbors. Moreover, edges are weighted by the street distance between parcels. In this generalized framework, however, the number of neighbors, as well as the particular kernel used to implement diffusion, are hyperparameters which need to be tuned.

The first formulation of our graph-based diffusion model starts each home at its Baseline Model predicted probability. Then at each step of diffusion, the node value is updated to be a weighted average of its own probability and that of its neighbors.[16] The results from this process can be visualized for pure performance as well as for an input in the stacking model below.

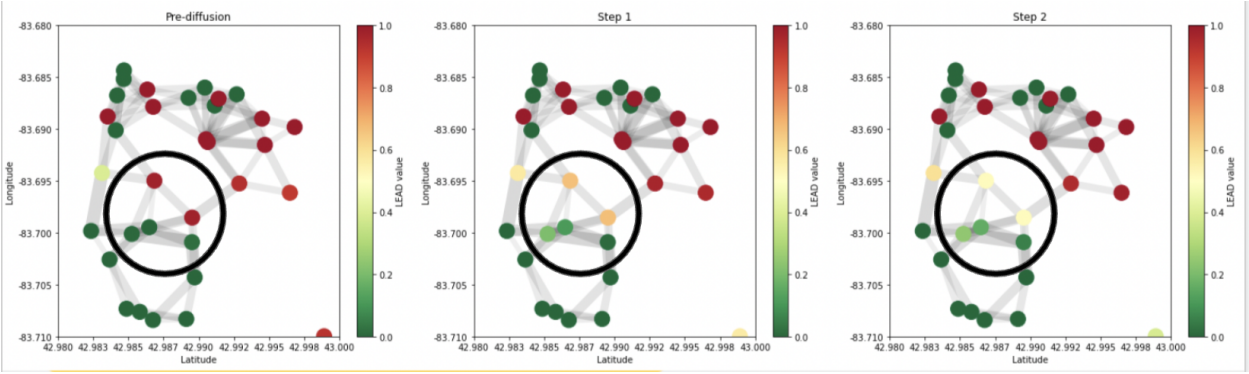


Figure 14: *We show what diffusion looks like step by step in a small neighborhood. The Pre-Diffusion plot shows the XGBoost Baseline Model's predicted probabilities. Then, we update each node's probability via a weighted average using the values of their neighbors. Notably, for the parcels circled in the geographic center of this plot, their neighbors on one side have mostly a low probability (green), and their neighbors on the other side have mostly a high probability (red). After two diffusion steps our probability estimates for this group of houses gets closer to 50%.*

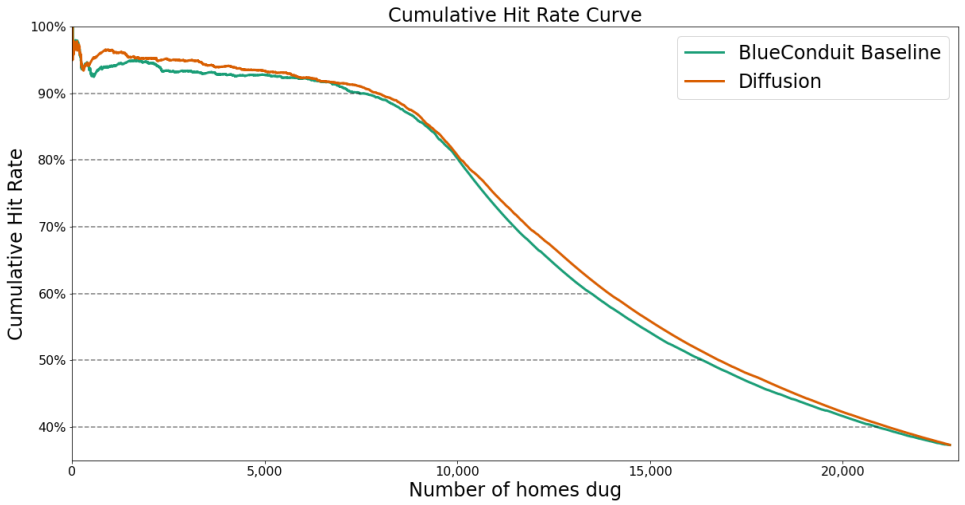
Below, we show the results for a single iteration of diffusion (i.e. one update step).[17] From these results, it is clear that under the partition-based evaluation metric, the diffusion alone suffices to improve the model. This is particularly important for the second half of the hit rate curve. These results are encouraging, and suggest that while the diffusion model may not improve the high-end predictions (i.e. those that the model is most confident over), it can help in areas of lower confidence, as

[16] As with the number of neighbors and the distance metric, the exact weighting of the self-probability and its neighbors will be tuned. Results will be presented in upcoming milestones.

[17] Figures for additional splits can be found in our GitHub repository. Please contact authors for access.

represented by the prediction probability. While the results from a single split of the 22 by 22 hex tiling are presented below, we demonstrate the results over additional splits in Appendix A. Under this tiling, each hexagon represents roughly 20 city blocks for Chicago or Manhattan.[18] In general, the diffusion models performed best when looking at smaller resolutions (i.e. a larger N x N), and performance was better when exploring geographically larger areas.[19] This is intuitive for all models -- when exploring large areas, the evaluation methodology proposed above is better able to approximate a true, unrestricted ordering. By deciding to investigate at the block or neighborhood level, there is more inefficiency in the process.

In Appendix B, we present the results of the hyperparameter tuning for the models using a customized loss metric, the area under the hit rate curve (where higher area indicates a better hit rate curve performance for longer). Ultimately, we selected the road distances, a linear distance weight, a self-weight of 0.6, and 50 neighbors. Our testing suggested that within localized hyperparameter regions, these models were not particularly sensitive to the hyperparameter choice. For example, results were comparable between 10 and 50 neighbors, whereas higher self-weights had significantly worse performance.



[18] A typical Chicago block is roughly 660 feet by 330 feet or roughly 217,000 square feet. The 22 x 22 hexagons are roughly 4,380,000 square feet. See https://www.chicago.gov/dam/city/depts/cdot/StreetandSitePlanDesignStandards407.pdf.

[19] As additional points of comparison, we have used 47 x 47 hexagon tiling, which corresponds to roughly 8 city blocks, and 99 x 99 hexagon tiling, which corresponds to roughly the block-level for Flint. Because these are evenly spaced, this may not accurately capture the Flint geographic structure but provides a helpful guide.

Figure 15: Results for diffusion for a single iteration of diffusion. Note that this relies on the 22 x 22 hexagon tiling, and represents a single split with 10% of the partitions included in the training set.

The second methodology for diffusion begins with only a set of labeled homes. From there, the network is evolved over many diffusion steps such that the probabilities of the test set(s) are changed over time to reflect the presence of lead in homes in the train set(s). In principle, this is similar to the Gaussian process. Rather than parameterize the effect of each lead home via a 2-dimensional Gaussian distribution, however, we would parameterize a graph and diffuse the information via that mechanism. One particular challenge for this model is that it is not clear how we would obtain prediction probabilities for the train set to be fed into a stacking model (described in Section E). Due to the time constraints of this project, we did not implement this method. We do, however, believe that it could be done through our ServiceLineDiffusion model class in our provided repository. To do so, we anticipate setting the train set to be fixed, true labels and beginning all test set homes at a baseline prior such as 0.5 and then running many iterations across the graph.

E. Graph Neural Network (GNN)

Our investigation into graph neural networks is motivated by the need to preserve BlueConduit's XGBoost *when it is working*. We have at our disposal a) the OpenStreetsMap road distance graph, B) our dataset of 301 features for each home, and C) the baseline XGboost probabilities. Our intuition is that the graph neural network paradigm has the following benefit: it can *learn* how to use nearby home features to inform how much we should spatially diffuse probabilities. As of milestones 2 & 3, however, the GNNs have been unable to surpass the XGBoost baseline.

In the Appendix, we describe the three different graph neural network architectures we have examined so far, and show the results for each so far by plotting their hit rate by partition.

F. Stacking

During model building, we noticed that the baseline XGBoost model had very confident predicted probabilities for lead (or not-lead) in many parcels. Our spatial models would often take these probabilities and "smooth them out" over nearby homes. In other words, confidence was distributed from "confident homes" (for

which the XGBoost model had predicted lead probabilities close to 0 or 1) to nearby "unconfident" homes (for which the XGBoost predicted probabilities were far from 0 and 1).

The upside of the above process is that the "unconfident" homes get to borrow information from nearby "confident" homes, ideally leading to more confident and accurate predictions. The downside is that the "confident" homes have their confidence diffused away from them, leading to less confident and potentially less accurate probabilities among them.
The above tradeoff motivated our use of **stacking**. By stacking models, we can create a model ensemble that learns to distinguish situations in which spatial information is helpful from situations in which spatial information is unhelpful. Specifically, the stacked model learns which particular sets of home features and XGBoost predicted probabilities indicate that a home's prediction would benefit from spatial information. Then, it only uses spatial information when predicting among those homes. Otherwise, it defaults to simply using the baseline XGBoost predicted probabilities.

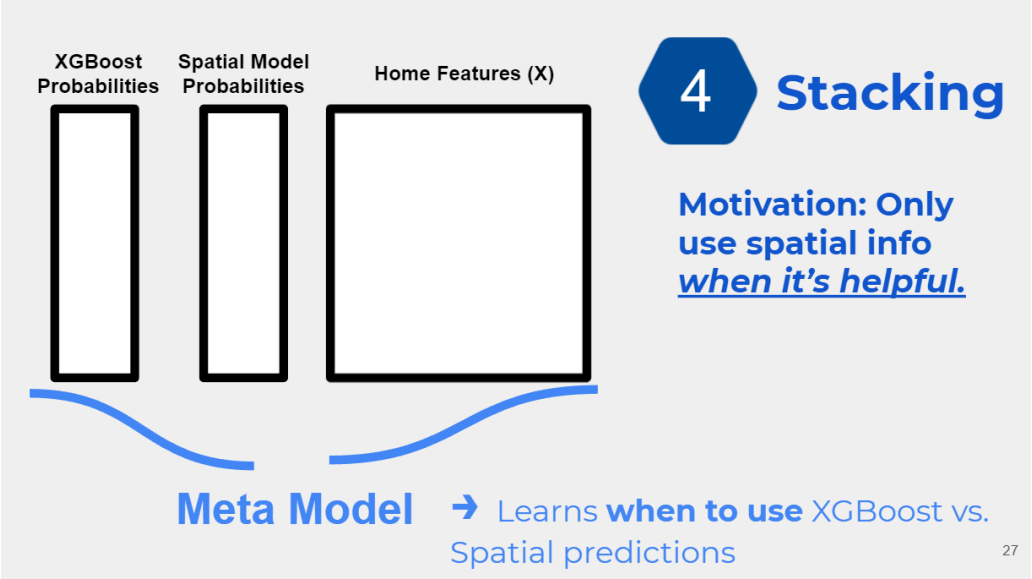


Figure 16: *Overview of meta-model ensembling process.*

As the above figure demonstrates, stacking works by learning a "meta model" over the XGBoost probabilities, spatial model probabilities, and home features. We implemented two different types of meta model in our stacking procedure:

- Tree-based models: decision tree and XGBoost
- Logistic regression

Our motivation for using tree-based meta models was the idea that such models could learn specific subsets of home features that indicate prioritizing spatial predictions over the baseline XGBoost predictions. Then, the model could make splits that use the spatial model probabilities in these cases. Our motivation for using logistic regression was that, empirically, simpler meta models (logistic regression, weighted averages, etc.) tend to perform better in most stacking tasks in machine learning.[20]

We trained our meta models using the XGBoost baseline probabilities, home features, and three spatial models: Diffusion, Gaussian Process (latitude, longitude, and year built), and Graph Neural Network (GraphSage). We also trained versions of these meta models that didn't include the home features and/or didn't include all of the different spatial models. Unfortunately, across all types of meta models, all combinations of spatial models, all train/test splits, and all hex tiling resolutions, stacking could not improve on the baseline model. The following hit rate curve is representative of the general results we saw from our stacking models:

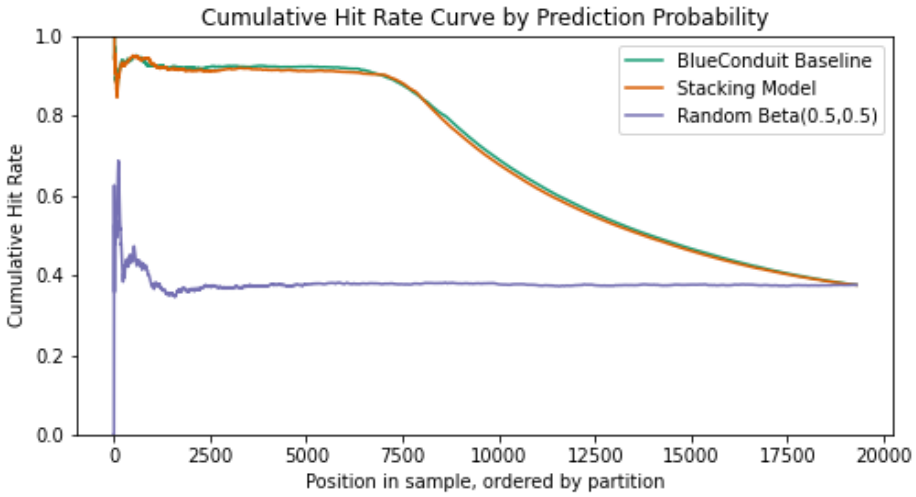


Figure 17: *Hit rate curve comparison of BlueConduit Baseline and Stacking Model*

Upon investigation, we found two main reasons why our stacking models failed to raise the hit rate curve above the baseline:

- For the tree-based meta models, splits were only being made on the XGBoost baseline probabilities. So, they simply used the following criteria: If XGBoost baseline probability > 0.5, predict lead. Otherwise, predict no lead.

[20] We spoke with Eagon (TF), who shared that tree-based meta models tend to overfit in Kaggle competitions and that simpler logistic regression or weighted average meta models tend to perform better on test data.

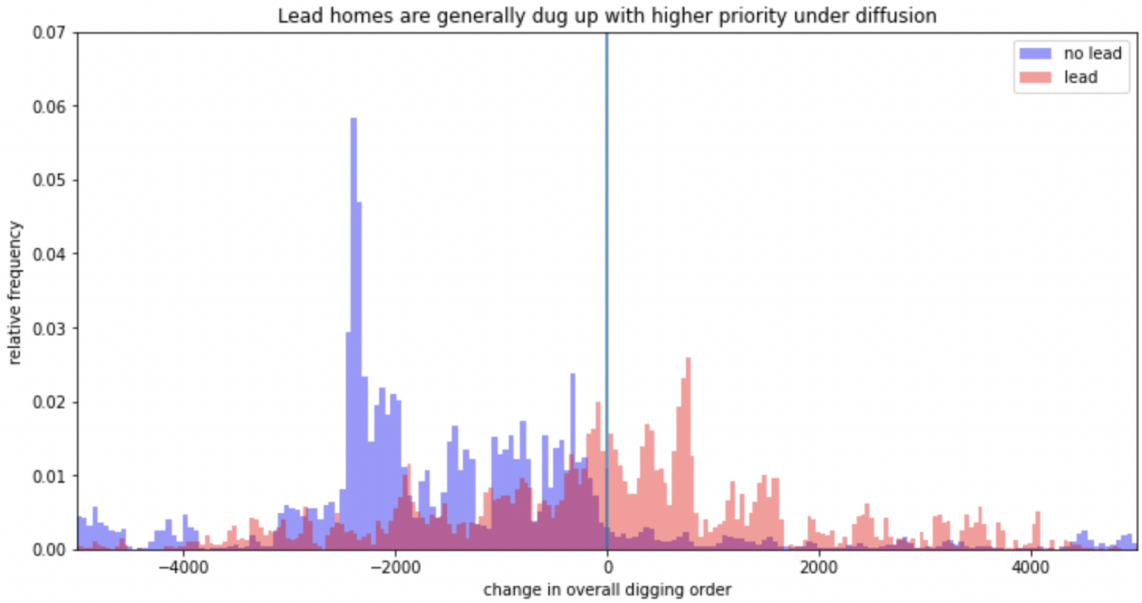
This result was consistent even after various attempts to give extra weight to the spatial model probabilities through transformations and hyperparameter tuning. We think this is due to the fact that these tree-based models use a binary cross entropy loss function - the same loss function used by the baseline XGBoost model. Since the baseline XGBoost model is already optimal according to this loss function, the tree-based meta models use the baseline for its splits in every situation.

- For the logistic regression meta model, some coefficient weight was given to the spatial model probabilities. So, unlike the tree-based models, the logistic regression meta model was using the spatial modeling probabilities. However, because the same coefficient weights were applied to every home regardless of its specific features, it had the same “smoothing” effect of the XGBoost baseline probabilities that we saw from our other spatial models. Overall, this produced a less optimal result.

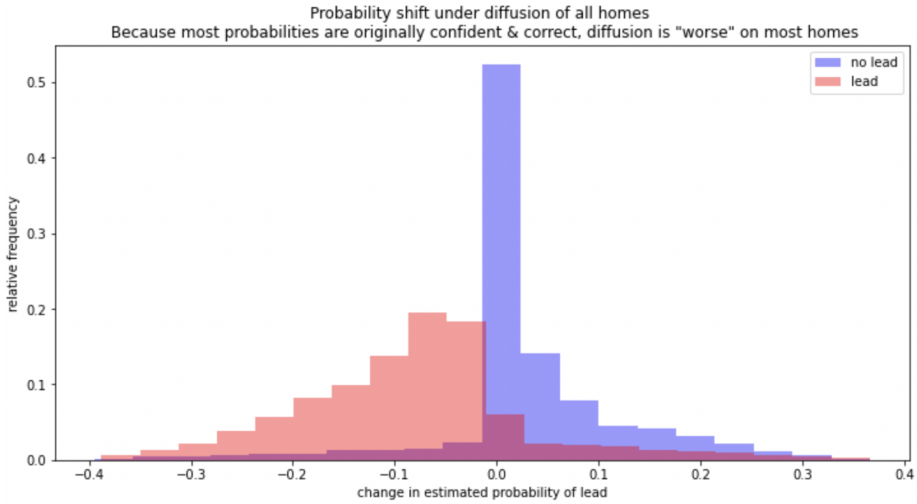
Discussion

Improvements from Diffusion

After running diffusion on the BlueConduit Baseline predictions, lead homes climbed 327 positions (on average) in the dig queue. Non-lead homes fell 195 positions, on average. The figure below shows the full distribution of dig order changes among lead and non-lead homes. It's clear that lead homes tended to climb in the dig order (positive values) and non-lead homes tended to fall (negative values).



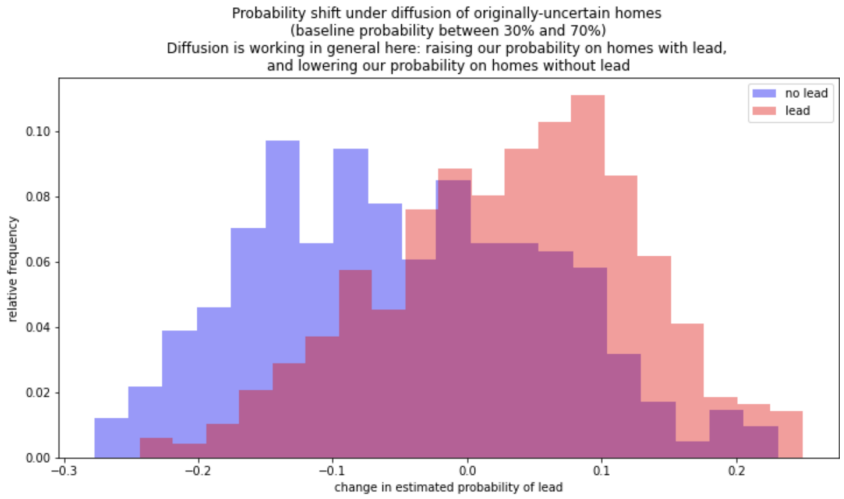
Exploring further, we found evidence of beneficial information sharing between homes. The figure below visualizes the changes in predicted lead probability (after diffusion) among all homes.



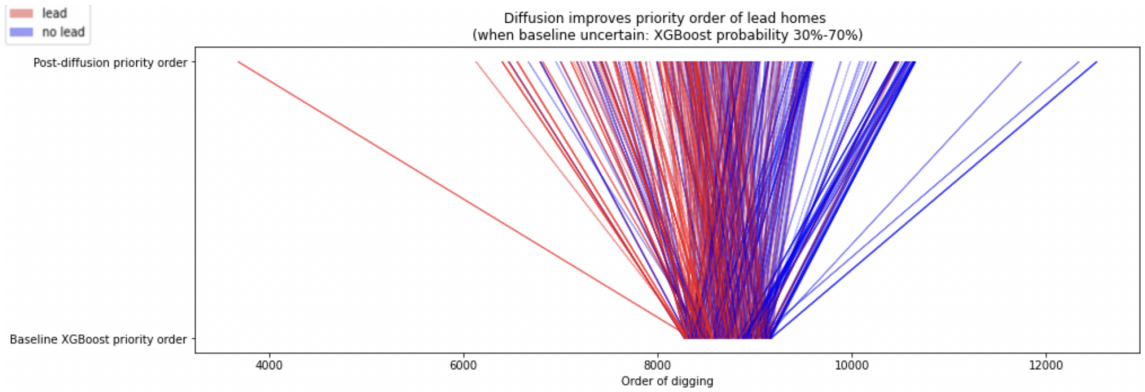
We see a seemingly counterproductive trend: diffusion tends to raise the lead probabilities of non-lead homes and lower the lead probabilities of lead homes. However, it is important to note that most homes were given highly certain and highly accurate predictions from the BlueConduit Baseline. In other words, most lead homes were given probabilities close to 1, and most non-lead homes were given probabilities close to 0. So, we'd expect diffusion to slightly smooth out these probabilities, regressing their extreme values closer to the mean. This means that many lead homes (baseline probabilities close to 1) had slight downwards shifts in lead probability and many non-lead homes (baseline probabilities close to 0) had slight upward shifts in lead probability.

Given this counterproductive trend, how is it possible that lead homes tended to climb in the dig queue and non-lead homes tended to fall? The key is to focus on highly uncertain homes. In the figure below, Panel A visualizes the change in lead probabilities, but only among homes that were given middling lead probability values (30% - 70%) by the BlueConduit Baseline. Here, we see a productive trend: diffusion increased the probabilities of lead homes and decreased the probabilities of non-lead homes.

A)



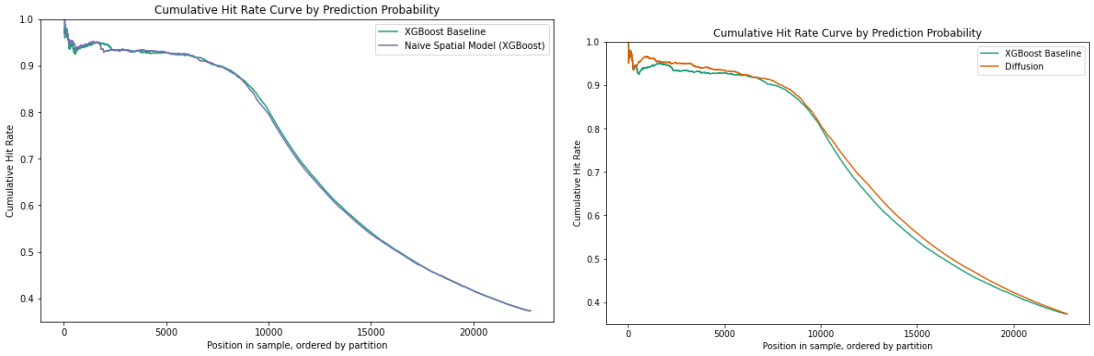
B)



Panel B shows the changes in dig order among a random sample of these same homes. The bottom of the graph represents their place in the BlueConduit Baseline dig queue. The top represents their place in the final diffusion dig queue. Because these homes were given uncertain prediction probabilities, they lie in the center of the BlueConduit Baseline dig order. However, we see that diffusion pulls lead homes higher in the dig order, and it pushes non-lead homes lower in the dig order.

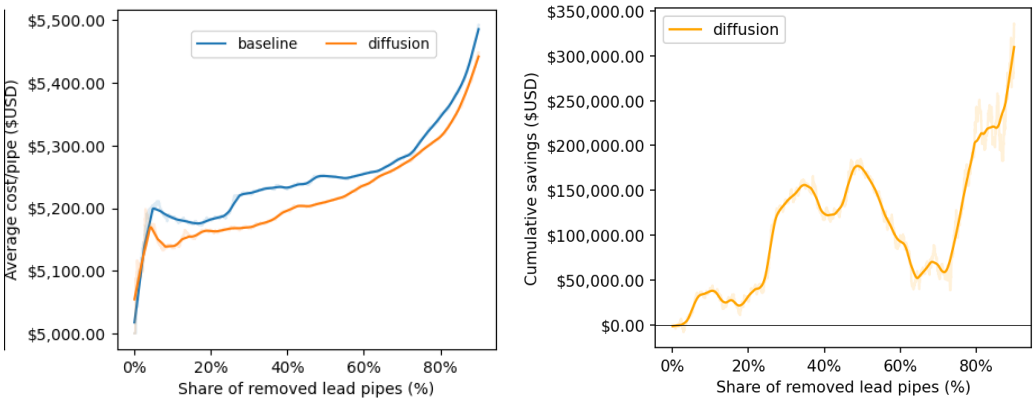
These results suggest that information is productively shared from certain homes to uncertain homes. We see evidence that uncertain lead homes (lead homes with middling probability values) tend to be located near lead homes with higher certainty (lead homes with probabilities close to 1). Diffusion allows these uncertain homes to borrow predictive strength from their neighbors, resulting in higher lead probabilities and a higher place in the dig queue. On the flip side, uncertain non-lead homes (non-lead homes with middling probability values) tend to be located near more certain non-lead homes (non-lead homes with probabilities

close to 0). Diffusion allows these homes to borrow predictive strength from their neighbors, resulting in lower lead probabilities and a lower place in the dig queue.



Importantly, the positive effects for uncertain homes outweigh the potential drawbacks for more certain homes (whose extreme probability values are regressed towards the mean). Above, we visualize two sets of hit rate curves. In the left panel, we compare the hit rates of the BlueConduit XGBoost Baseline and the BlueConduit XGBoost Baseline with latitude/longitude included as predictors (labeled: “Naive Spatial Model”). We see that naively including latitude and longitude as predictors produces no noticeable improvement in hit rates. In the right panel, we compare the hit rates of our diffusion model against the same BlueConduit XGBoost Baseline model. Here, we see evidence that the diffusion model has a higher hit rate in both the front half and the back half of the dig queue.

Although the raw differences in hit rates appear small, the estimated differences in savings for the city are quite large. The figure below shows estimates of these savings.



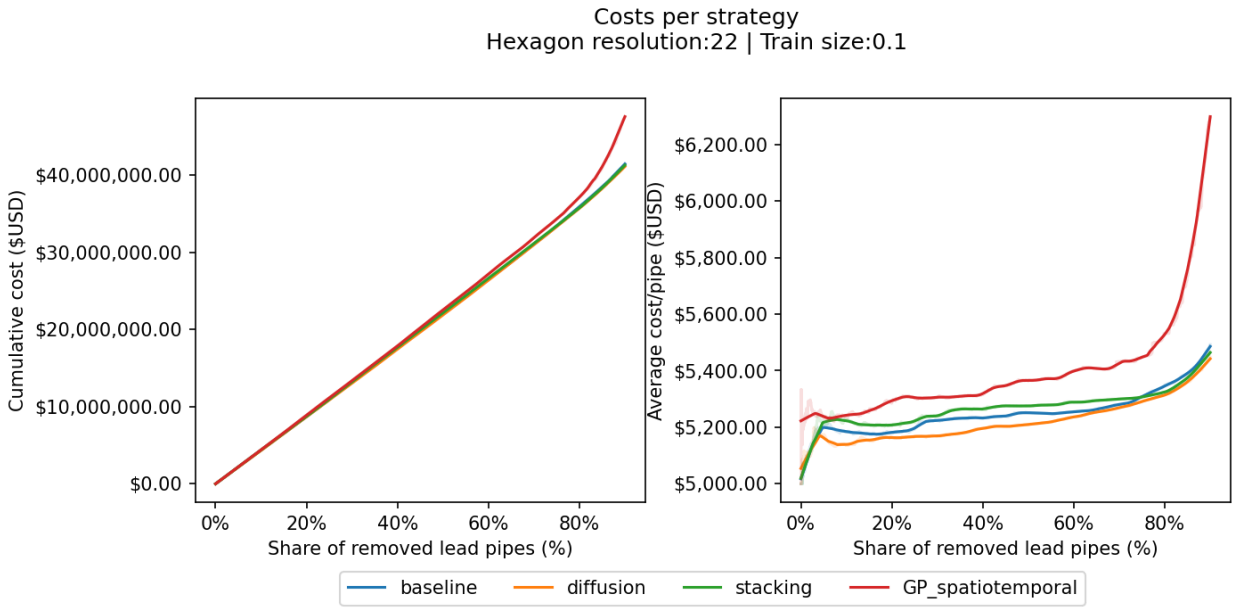
Because diffusion tends to push lead pipes forward in the dig order (and pull non-lead pipes down), it takes fewer total digs to find and replace most of the lead

pipes in the city. This translates into monetary savings. As visualized above, diffusion clearly lowers the average cost of replacement throughout the digging process. By the time that 90% of lead pipes are dug, diffusion saves more than $300,000 total for the city (relative to the BlueConduit Baseline model). These savings are primarily driven by preventing wasteful digs of non-lead homes.

In the following section, we dive deeper into our cost savings results.

Cost Results

We now visualize different costs and savings curves to assess the monetary impact and improvements of our proposed models compared to the Blue Conduit baseline model. The cost curves depict the cumulative and average costs of removing the p% share of lead pipes per strategy. These costs incorporate all the unnecessary digs before reaching the given number or percentage of lead pipes removal. The average cost is calculated as the cumulated costs divided by the number of lead pipes removed. For the calculation, we consider contractors are paid $5,000 to replace a service line and $3,000 for an excavation that does not yield a replacement[21].



[21] These costs were calculated following Webb et al. (2019) methodology. See Jared Webb, Jacob Abernathy, and Eric Schwartz, "Getting the Lead Out: Data Science and Water Service Lines in Flint," *Bloomberg Data Exchange for Good*, 2019. Available at: https://storage.googleapis.com/flint-storage-bucket/d4gx_2019%20(2).pdf.

Figure 18: *Cumulative and average costs of removing p% of lead pipes.*

From these cost curves, we can calculate the savings over the baseline strategy as the costs difference between the baseline strategy and the given strategy:

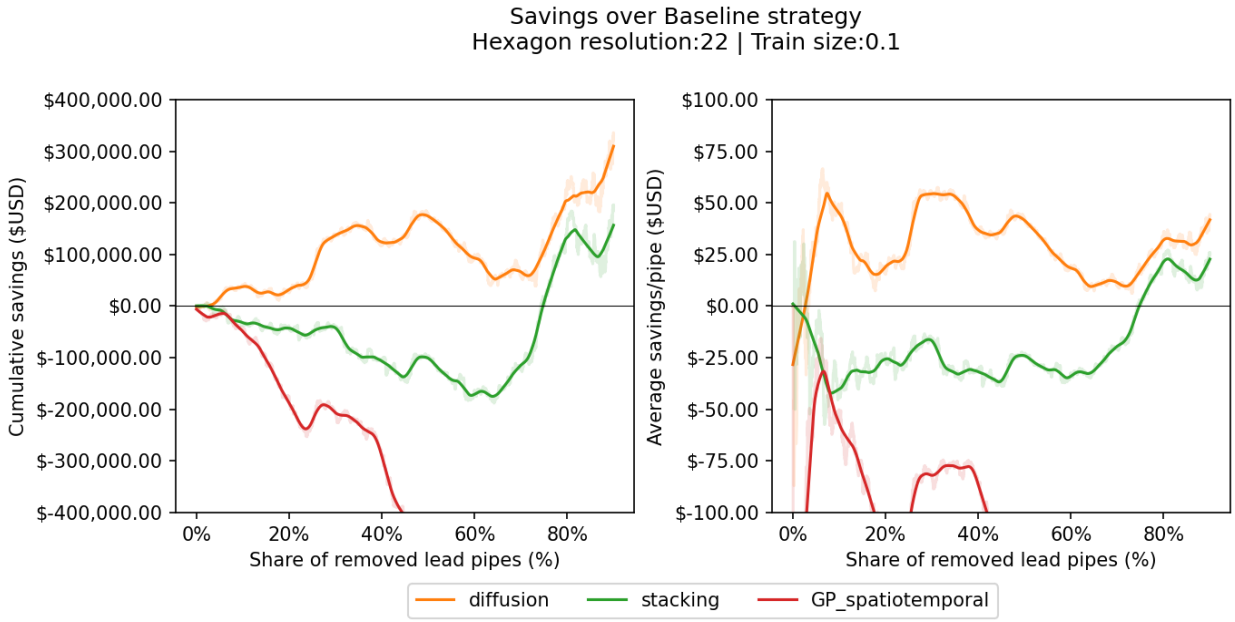


Figure 19: *Cumulative and average savings with respect to the baseline of removing p% of lead pipes. Resolution=22 and train size=0.1*

The diffusion model consistently has positive savings over the baseline model, despite showing significant fluctuation in the savings throughout the leading removal process. For the case, we observe savings as much as $50/lead pipe and cumulative savings for as much as $300,000.00. This behaviour repeats across the majority of train sizes and hexagon resolutions. Note that to improve readability we have incorporated only the “diffusion”, “stacking” and “GP_spatiotemporal” as they were highests performers in a savings regard across all resolution and train sizes scenarios.

Now we analyze the effect of the train size over the savings. Larger training sizes are associated with less variability in the curves but also minor improvement in savings, as depicted in the following figure:

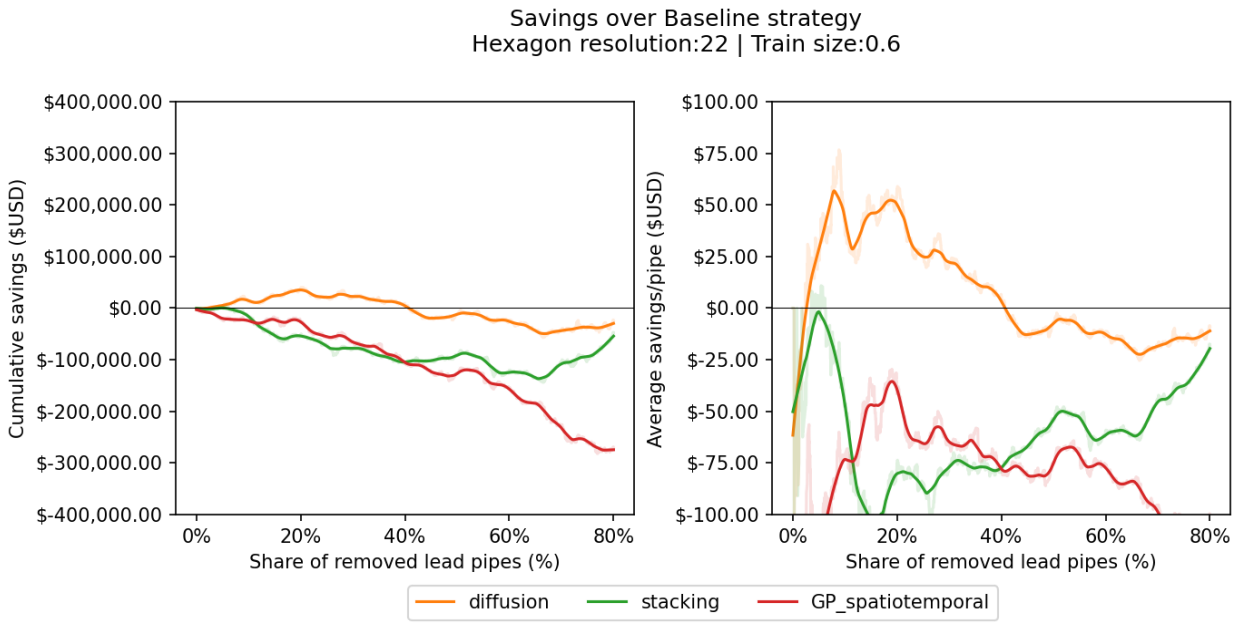


Figure 19: *Cumulative and average savings with respect to the baseline of removing p% of lead pipes. Resolution=22 and train size=0.6*

Presumably, there's less space for improvement with bigger train sizes as the baseline model improves its performance on the unobserved samples, and thereby it becomes increasingly harder to beat it.

Finally, lower resolutions are associated with significant improvements over the first samples but fade away faster than higher resolutions. Lower resolutions yield more parsimonious smoothing by increasing the neighbour scope but lose the ability to define lead pipes based on the immediate neighbours. Diffusion leverages the locality component by selecting partitions/hexagons with a high predicted number of leads. Since higher resolutions have only a few partitions, most improvement relies on the parcel ordering; diffusion will select the most promising partitions and afterwards will have little impact on the further savings improvement. This description is illustrated in the following Figure for the lowest hexagon resolution (5):

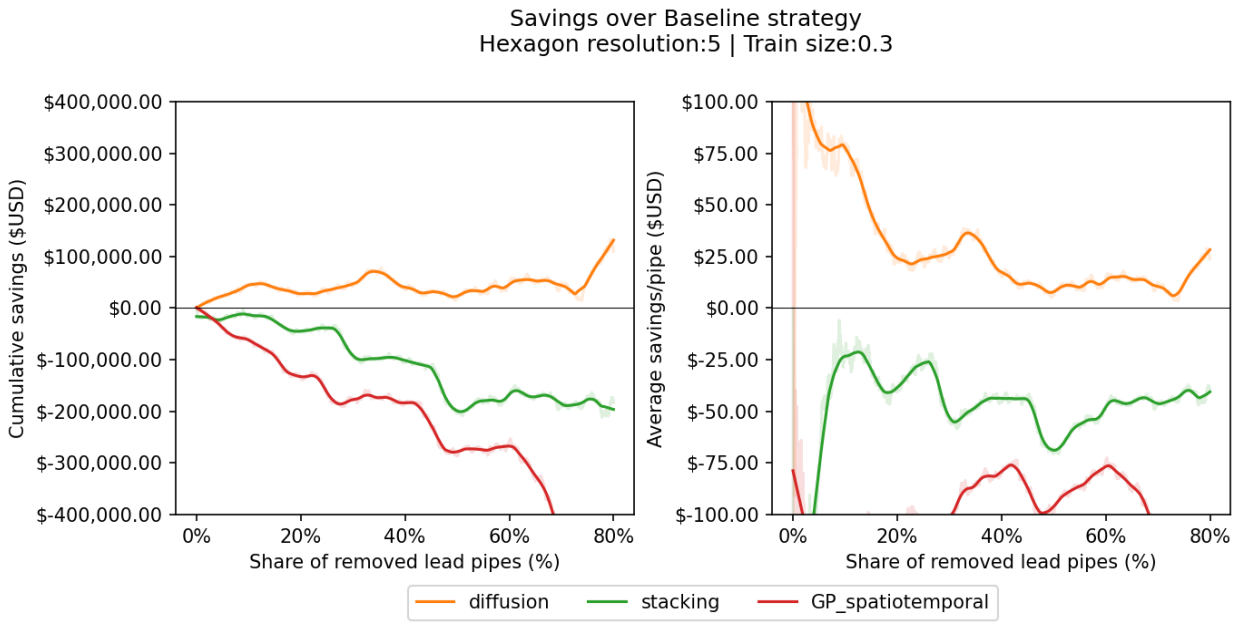


Figure 20: *Cumulative and average savings with respect to the baseline of removing p% of lead pipes. Resolution=5 and train size=0.3*

What is the diffusion mechanism?

One downside of graph-based diffusion relative to a more parametric approach is that there are few model internals which can be consulted to understand the importance of features in this model relative to the baseline. Some insight, however, can be gained from the model hyperparameter tuning. For example, the best-performing models tended to place more than half of the weight on the baseline (i.e. in the weighted average, this was typically a weight of 0.6 or 0.7 relative to the weighted average neighbor value). This suggests that even with diffusion, optimal performance heavily depends on a well-performing baseline model. Moreover, it is instructive to consider that the best performance did *not* come from the full weight being placed on the baseline which concurrently highlights that the smoothing effect provided by diffusion is important.

This is bolstered by the fact that the number of neighbors selected was typically in the range of 50-100 and that the road distances aided the model more than air distance. Beyond the value of the baseline, that result suggests that broad neighborhoods may be more important than a parcel's immediate neighbors, and that there are strong neighborhood effects. If direct proximity was more important, we would expect to see a smaller number of optimal neighbors and air distance outperform. With the 99 x 99 partitioning, the median partition has only 7 parcels

compared to 107 for the 22 x 22 partitioning. Thus, the scale of the optimal neighborhood tends to be at a more coarse neighborhood definition.[22]

We undertake two experiments in this section to better understand the performance gains. First, we perform a sort of "ablation study" using the partition ordering in construction of the hit rate curve to determine whether diffusion allows for better selection of neighborhoods during exploration or better ordering of parcels within each neighborhood investigated.[23] From this experiment, we conclude that much of the diffusion performance improvement is the result of investigating low-baseline probability parcels earlier on during the process. Because the algorithm to evaluate the performance relies on "sweeps" through the city with different probability bands, by moving low-baseline probability parcels into "earlier" bands, these homes are investigated relatively more quickly.

Ablation Study

To perform the ablation study, we recognize that under our diffusion algorithm, there are two places where the predicted probabilities are used: ordering the partitions and ordering the parcels within a partition. We attempt to isolate the effects of diffusion by manipulating these pairs. Our first comparison is between the "All Baseline" method and a method which uses the diffusion predictions to order the partitions but the baseline probabilities to decide which parcels to investigate. More specifically, given a threshold, we calculate the total number of parcels with a diffusion prediction probability greater than the threshold, which is used to order partitions. Once that order has been determined, however, the model investigates all homes with a *baseline* prediction above a certain threshold. Intuitively, if the diffusion primarily acts via better neighborhood-level prediction, we would expect to see substantial performance gains. All plots in the first section correspond to the 22 x 22 partition and the first split used. Results are similar for all splits, but not necessarily for all partitioning. Below we highlight the 99 x 99 partition, which features slightly different results.

[22] The 47 x 47 partitioning results in a median parcel count of 27. Mean parcel counts are similar but slightly larger, indicating that the distribution is somewhat left-skewed.

[23] Ablation studies, which come from neuroscience, deliberately restrict performance in some way to gain insight into the inner mechanisms of a model. These studies have become increasingly popular to analyze deep neural network models. See e.g. Richard Meyes et al., "Ablation Studies in Artificial Neural Networks," *arXiv preprint* (2019), available at https://arxiv.org/pdf/1901.08644.pdf.

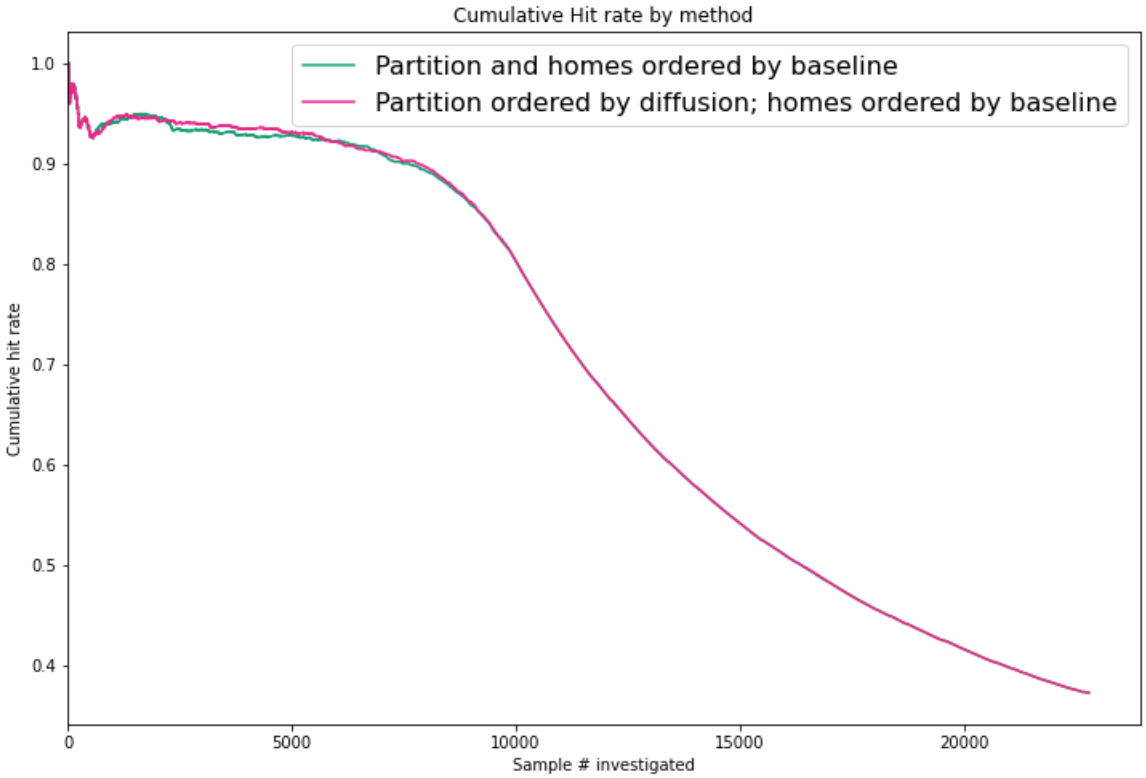


Figure 21: *Comparison of baseline-only and partition-diffusion configurations.*

From Figure 21, it is clear that switching to use diffusion for only the neighborhoods does not have a large effect. This is sensible given that each partition has a median of 107 parcels. While diffusion may make marginal differences, it is likely not causing dramatic reorderings of partitions, and those

that are reordered are likely locally similar (i.e. a partition with 10 parcels predicted and 11 are not likely to produce large jumps in performance).

Next, we consider the inverse: partitions ordered by the baseline but homes ordered by diffusion. Notably this can pick up two effects, ordering better at the partition-sweep level as well as at the unrestricted partition level. As can be seen at the right, the improvement in performance is pronounced. From this, we conclude that diffusion has a significant effect on the ordering of individual parcels.

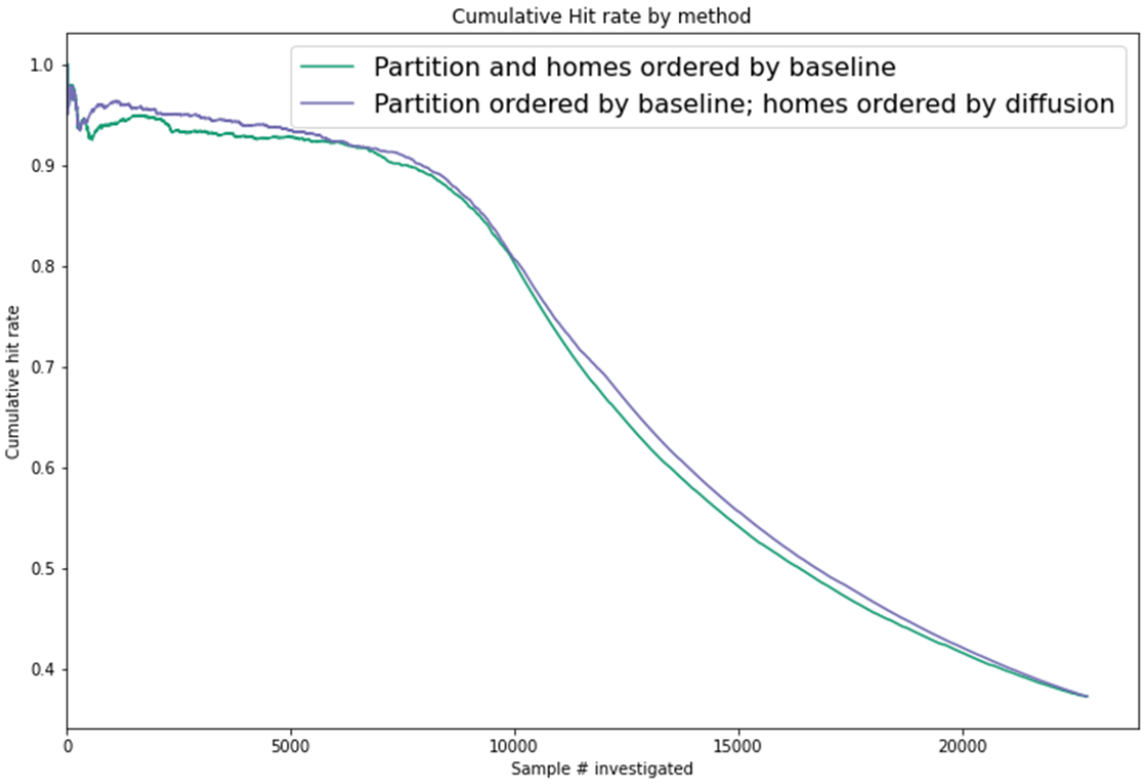


Figure 22: *Comparison of baseline-only and home-diffusion configurations.*

Finally, we investigate whether using diffusion in both facets increases performance by an additional margin. We find that for the 22 x 22 resolution, there is little additional effect from using diffusion to order the partitions as well as the parcels within a partition.

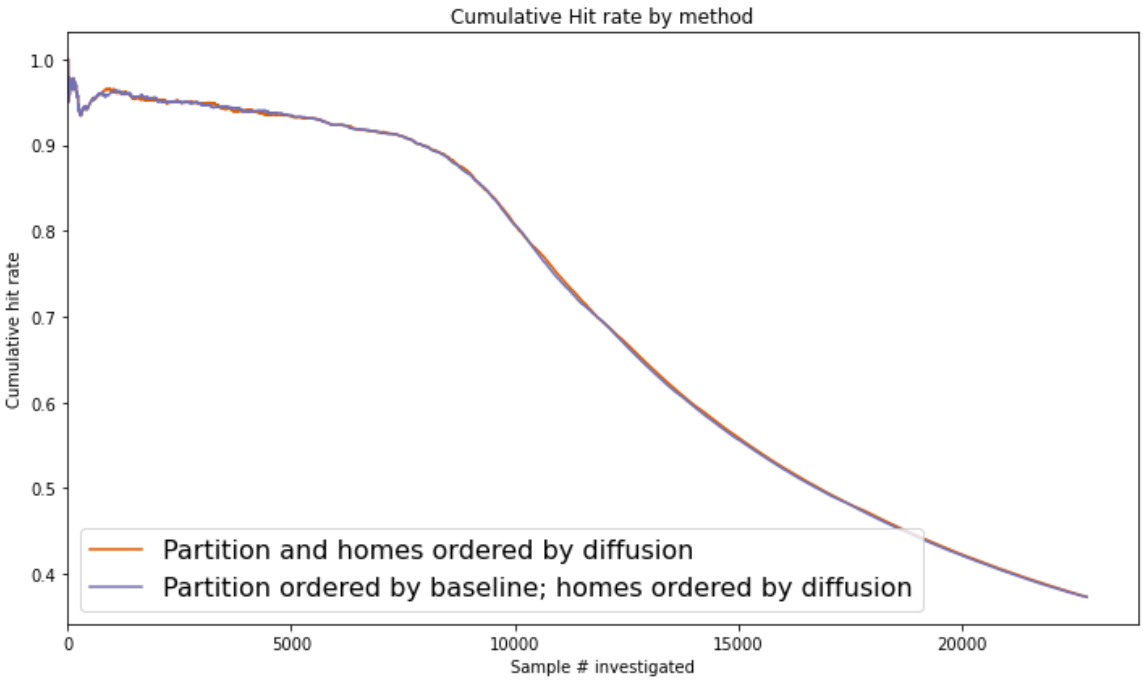


Figure 23: *Comparison of home-diffusion and diffusion-only configurations.*

After viewing this plot, we explore two additional directions. First, we consider the predicted probabilities of the diffusion model relative to the baseline. This illustrates that the effect of diffusion is primarily through digging more effectively within each sweep. In the plot below, we highlight the median prediction probability of the homes dug within each partition. Unlike the parcel-ordering, this will not be smooth. Rather, it proceeds in rough "bands", highlighting that some partitions are pursued before others and thus a home with a predicted probability of 0.9 may be investigated after one with a predicted probability of lead of 0.99 depending on the ordering of the partitions that they belong to. Here, we see that there are material differences not just on the average probabilities but on the total number of parcels investigated within each sweep.

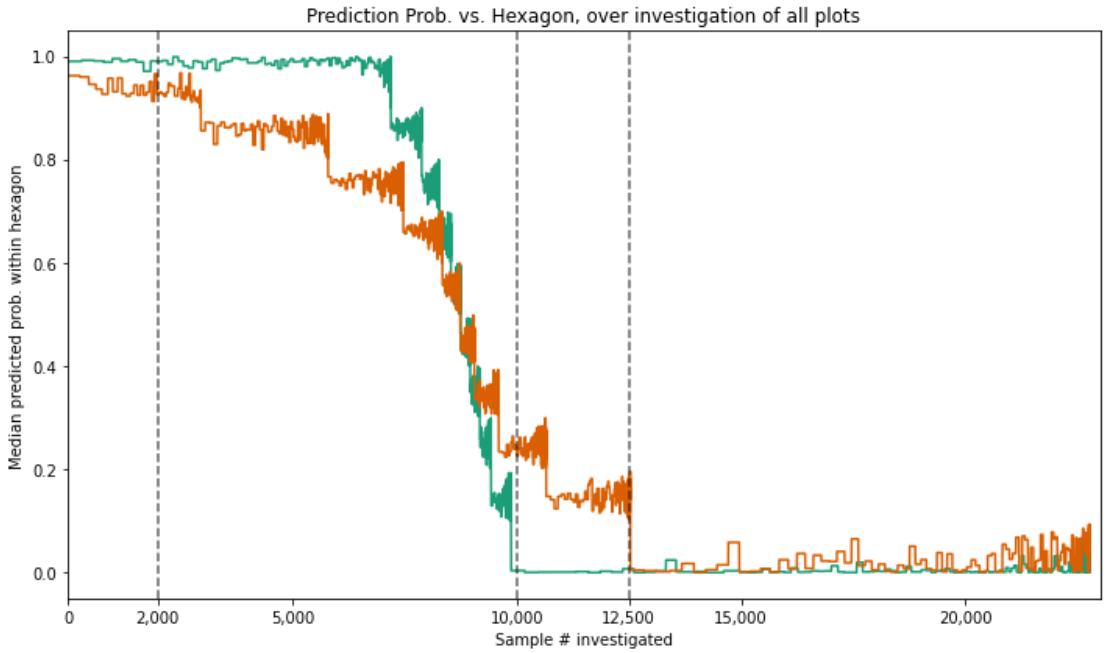


Figure 24: *Comparison of prediction probabilities between baseline only (green) and diffusion-only (orange) models.*

The dotted lines on this plot help visualize the results further. From the plots of the hit rate curve, there are roughly two areas where diffusion improves performance: early on, beginning at roughly the 2000th parcel investigated and once again after roughly 10,000 parcels have been investigated. For the predicted probabilities, this corresponds to roughly fewer parcels being investigated in the [0.9, 1.0] sweep and more in the [0.2, 0.3] and [0.1, 0.2] sweeps. Because of these results, we conclude that the effects of diffusion are strongest in adjusting which homes fall into each partition-sweep pair. For example, consider a home with a baseline prediction of 0.1 and which has neighbors predominantly having prediction probabilities in the [0.9, 1.0] range. Even if this home has not shifted relative position within its partition, we would expect that it would be investigated earlier. In that case, it is neither the ordering of partition within a sweep nor the relative ordering of homes within a partition which is important, but rather *which homes* are investigated within each sweep. Finally, we consider a more fine-grained partitioning, specifically the 99 x 99 partitioning in which each partition roughly corresponds to a city block. Under these conditions, the median partition has only 7 parcels.

Unsurprisingly, using diffusion for ordering partitions and parcels leads to substantial performance improvements for low-probability parcels versus using diffusion for the parcel-ordering only. When the partitioning is done at such a fine level, diffusion can help across adjacent blocks, achieving a better ordering.

Moreover, because each partition has so few parcels, we would expect that changes to a single parcel would be likely to change the entire ordering. This leads to an additional insight: when cities wish to investigate at a fine-grained level, the ordering of partitions, particularly in lower-expected lead neighborhoods and/or sweeps, is a more important determinant of outcomes. In fact, it a strategy wherein cities investigate whole partitions at this level could meet similar efficiency due to the fact that each would be expected to be visited many times for a small number of investigations at each sweep.

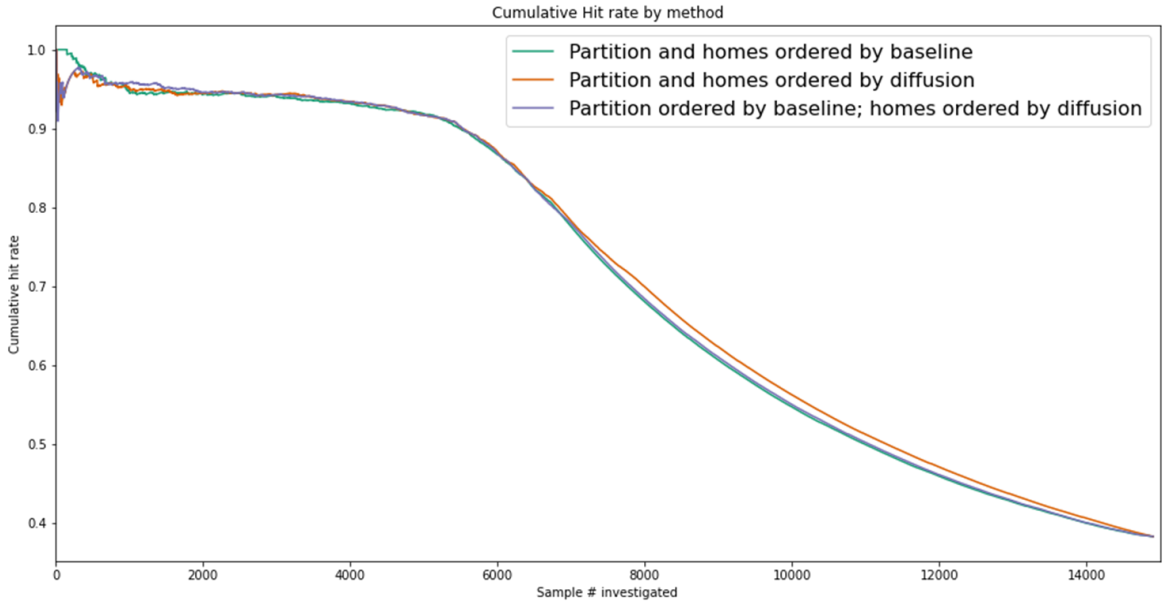


Figure 25: *Comparison of ablation study configurations for 99 x 99 partitioning of Flint.*

Regularization

An additional question that arises from these considerations is whether the diffusion improvements demonstrated above come from a form of regularization. Importantly, all of our modeling to date has featured only 10% of the partitions included in the training dataset. While this assumption reflects an early stage lead removal process, the baseline models may badly overfit to the particular parcel characteristics sampled. As a thought experiment, we consider the world where all neighborhoods sampled have complete information on the age of the housing stock. If there are some partitions where this information is not well maintained, the model may struggle by learning the intricacies of the years of only a few small city blocks.

We have analyzed the effect of two types of regularization on the baseline XGBoost model by varying the strength of L1 and L2 regularization, respectively denoted α

and λ.[24] For each parameter, we tested increasing strength of regularization and cross-validated via mean log loss to select a hyperparameter. Below is a representation of the baseline model, diffusion model, and L1-regularized baseline. As we see, the diffusion and regularized models are similar, highlighting the connection between the two. Interestingly, the diffusion model still tends to outperform the regularized XGBoost.

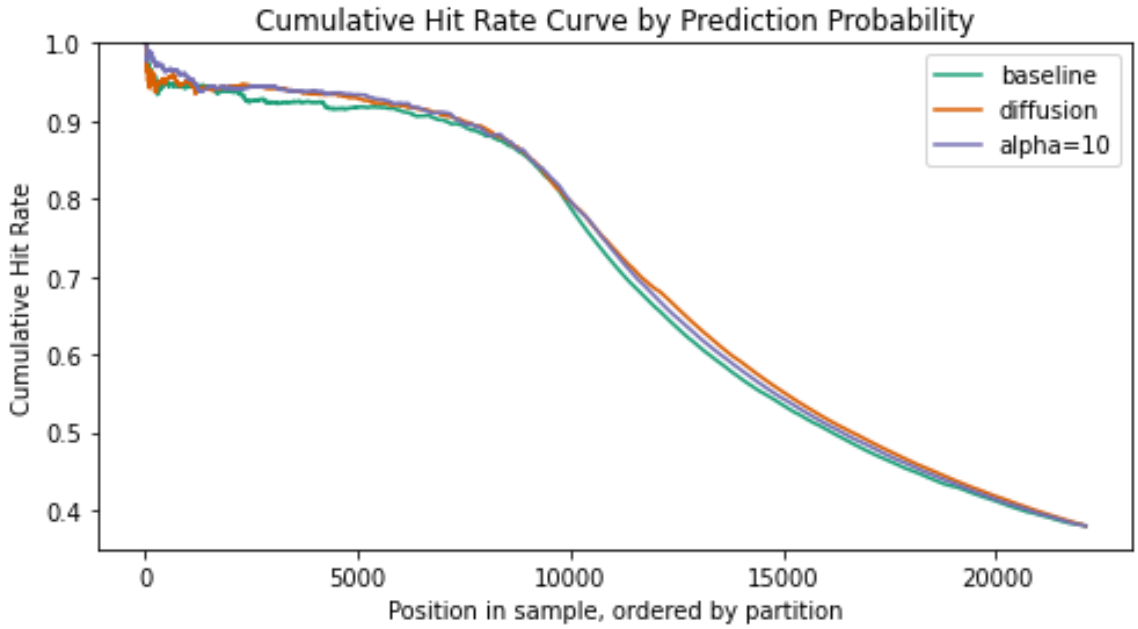


Figure 26: *Hit rate curves for BlueConduit Baseline, Diffusion, and L1-Regularized Baseline, with parameter selected via cross-validation (alpha = 10).*

Below we highlight the feature importances from the regularized XGBoost model. This confirms the exploratory data analysis by showing that Year Built is still a predominant feature in determining the outcome. Additionally, we observe that SL_TYPE_1, corresponding to the historical record, is far and away the most important predictor. Once again, this is intuitive, as we would expect that the historical records provide a strong signal even if they are sometimes incorrect and often incomplete.

[24] See Tianqi Chen and Carlos Guestrin (2016), "XGBoost: A Scalable Tree Boosting System," 2016, available at https://arxiv.org/pdf/1603.02754.pdf, and the XGBoost Documentation, available at https://xgboost.readthedocs.io/en/latest/tutorials/model.html.

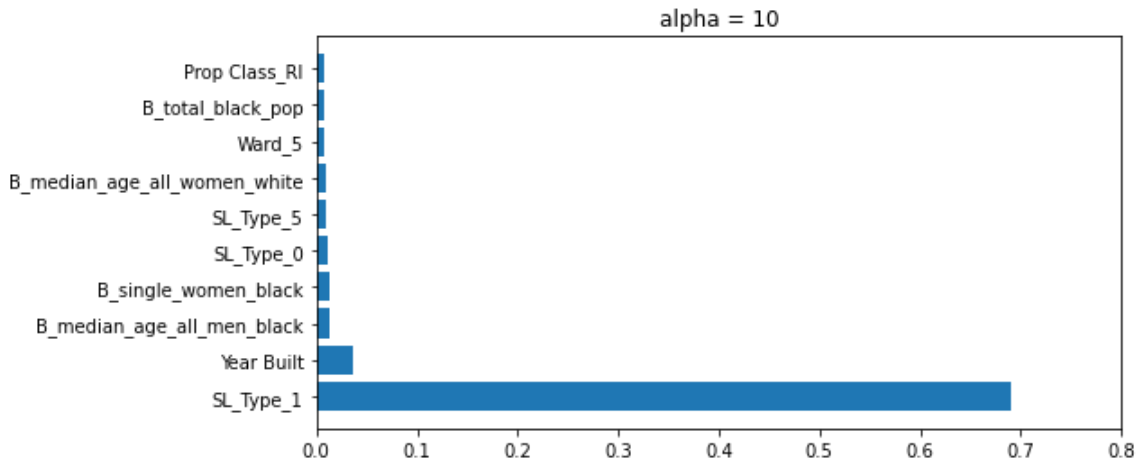


Figure 27: *Feature importance plot for L1-Regularized Baseline model with cross-validation selected regularizations strength.*

As a check on the sparse regularization presented above, we also consider the effects of various strengths of ƛ. As with all L2 regularization, we expect that more features will be included in the feature importance but with less substantial importances. From the plots below, this is exactly the pattern we observe. Notably, the hit rate curve for the L2 regularized model once again appears to split the difference between the baseline and diffusion models for samples later on in the ordering, but it is difficult to make a direct comparison between the two.

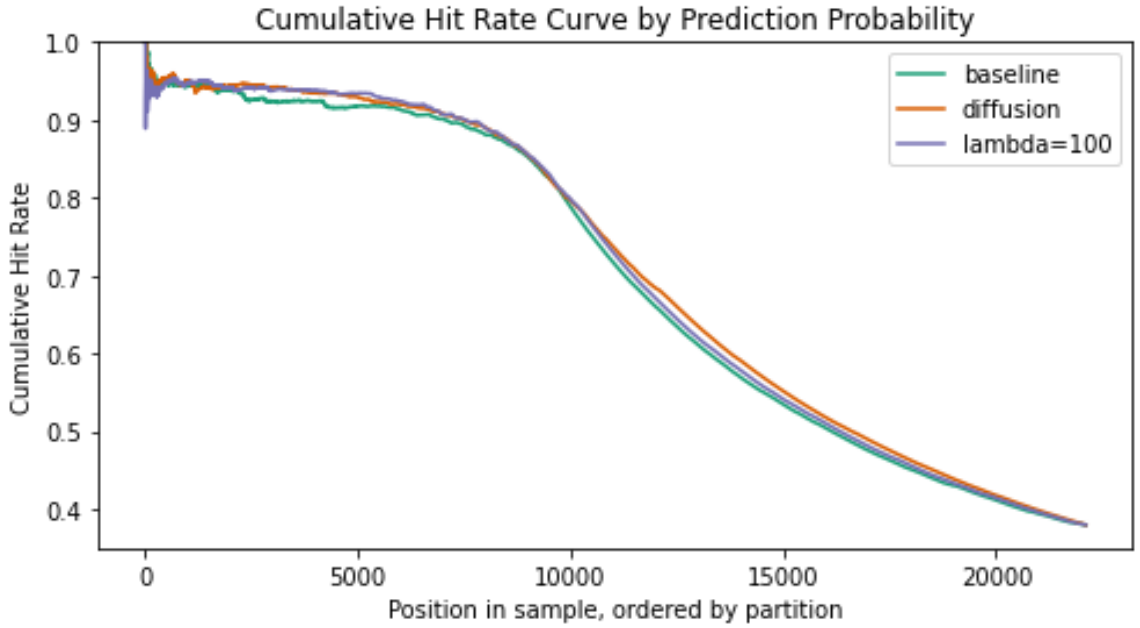


Figure 28: *Hit rate curves for BlueConduit Baseline, Diffusion, and L2-Regularized Baseline, with parameter selected via cross-validation (lambda = 100).*

At the 22 x 22 resolution, each method had similar results from the log loss metric via cross validation, confirming the similar results of each. Indeed, while log loss is not our primary metric of consideration, the reported log loss for each regularized XGBoost model was roughly 0.26 whereas the optimal configuration of hyperparameters produced a log loss of approximately 0.24. Both of these compare to the baseline log loss of roughly 0.33.

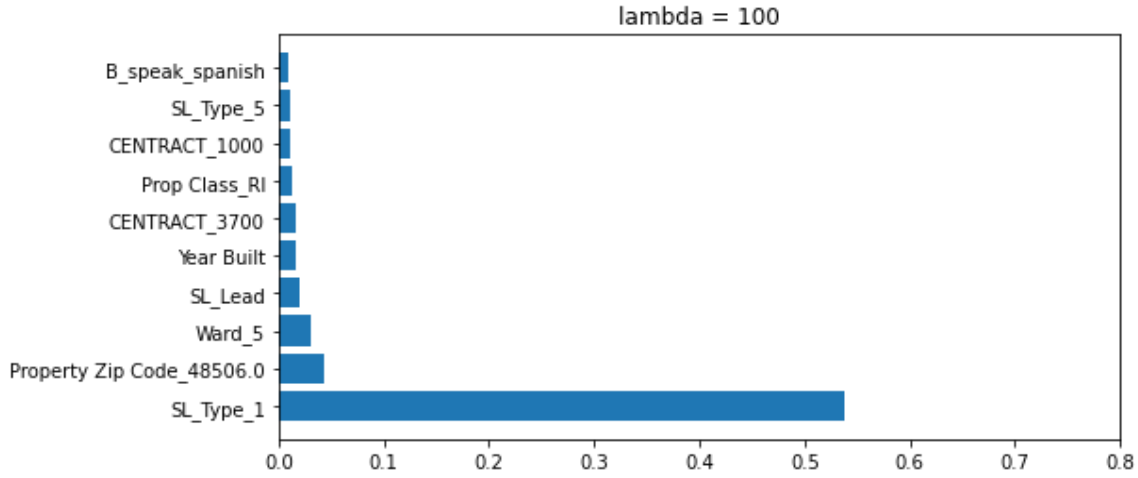
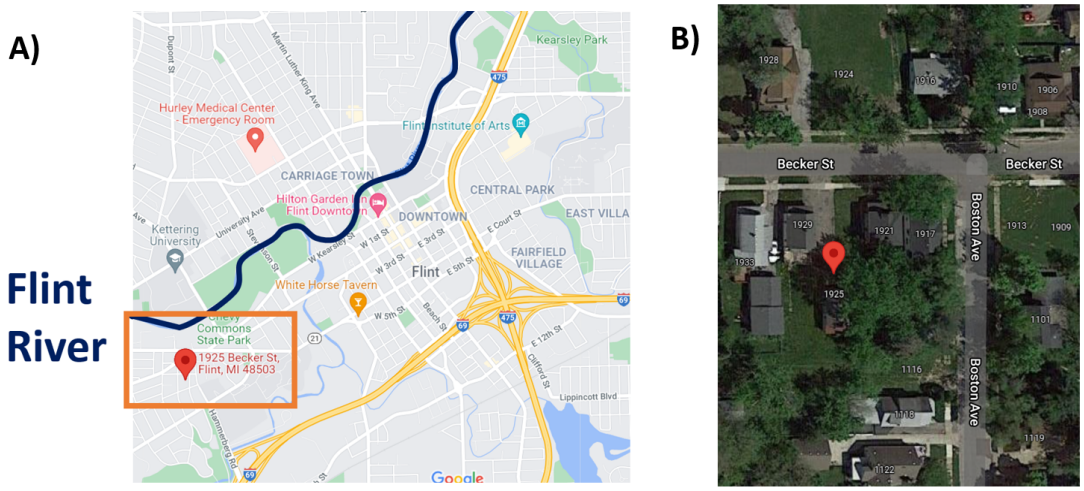


Figure 29: *Feature importance plot for L1-Regularized Baseline model with cross-validation selected regularizations strength.*

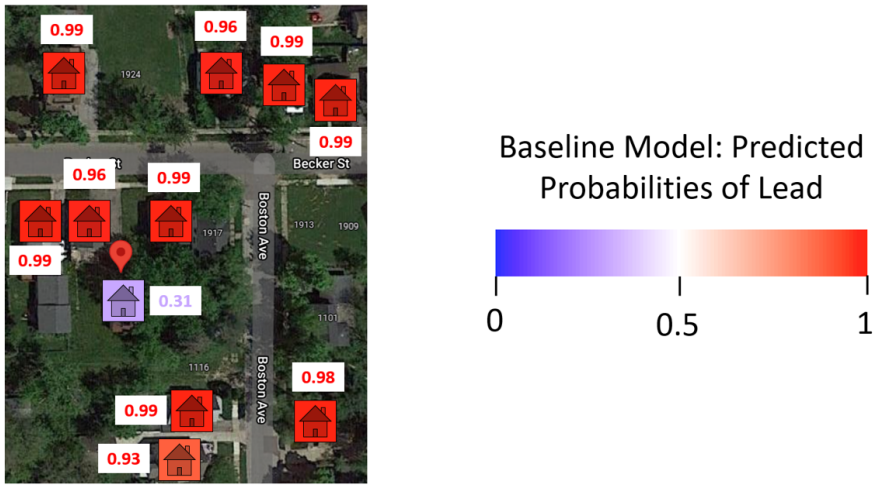
Case Study: 1925 Becker St.

Let's walk through an example to motivate diffusion and demonstrate how we used it.



Panel A in the above figure is a map of Flint. In 2014, the city switched its water source to the Flint River, highlighted in dark blue. This decision marked the beginning of the Flint Water Crisis, as the Flint River water contained chemicals that corroded lead pipes in the city. Close to the Flint River is a residential neighborhood on the west side of the city, marked by the orange box. This neighborhood had a high density of lead service lines. A particular home – 1925 Becker St. – is located within this neighborhood.

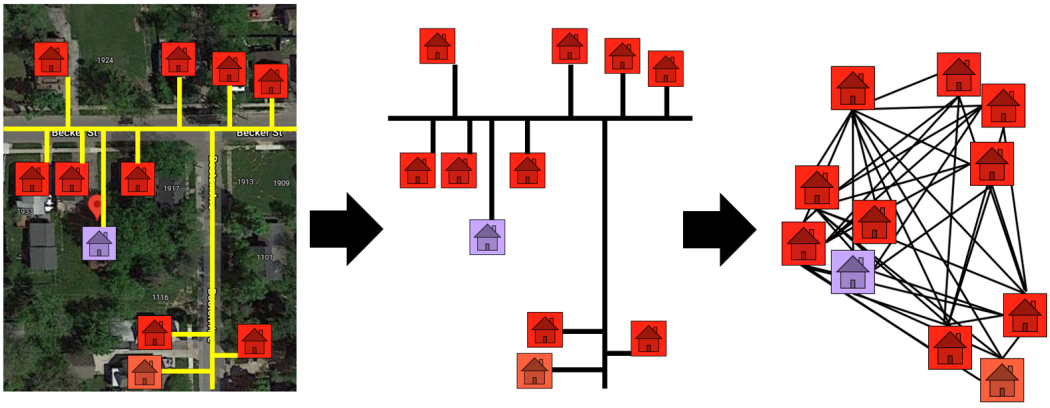
Panel B zooms in on 1925 Becker St. (marked by the red indicator) and its immediate neighbors. Because the pipes in Flint were all dug up in the years following the Water Crisis, we know the ground truth: every home in the picture had lead pipes. So, if this neighborhood was in a test set (i.e. not yet dug), an ideal model would give high probabilities of lead to every home. The figure below shows the lead prediction probabilities given to each home by the BlueConduit Baseline model.



The BlueConduit Baseline gave each home a high probability of lead (>90%), except for one: 1925 Becker St (31%). Upon inspection, it appears that this home had a peculiar feature: its city record indicated that it had copper (safe) pipes. Flint's pipe records are often unreliable. However, records that explicitly indicate copper pipes are usually accurate.The BlueConduit Baseline model (an XGBoost model trained on non-spatial features of the home) keyed in on this feature and was "fooled." The model gave this home a lower probability of lead, placing it at position 9,136 in the queue of homes to dig. If digging resources were limited, teams may have never reached this home. Instead, they may have dug homes higher in the queue that didn't, in fact, have lead pipes. How could we prevent this result?

As we mentioned above, because homes in the same neighborhood are often built by the same developer, we believe that close neighbors should have similar lead probabilities. Because the BlueConduit Baseline doesn't use spatial features, it cannot share information between proximate neighbors. However, this type of information sharing can be achieved through diffusion.

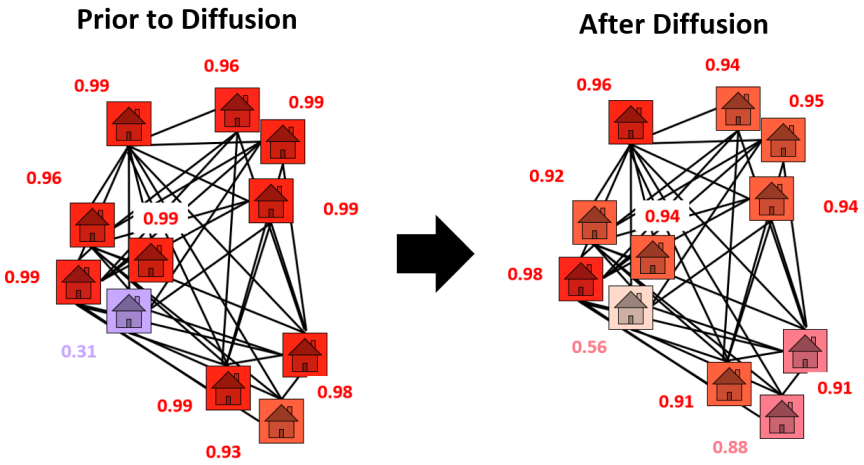
To set up diffusion, we first had to model and build a graph of the distances between homes. Our graph-building process is demonstrated by the figure below. First, we used Open Street Maps to find the street distances (Manhattan distances) between homes. We opted to use street distances instead of Haversine distances ("as the crow flies" distances) because streets encode some of the shared development and infrastructure between homes. Housing developments proceed block-by-block rather than area-by-area. So, two homes with adjacent backyards (small Haversine distance) may have been built by different developers – especially if they aren't connected by a shared road. In addition, pipes are usually built underneath roadways. So, street distances can also encode shared water mains.



After obtaining the road distances, we created a graph connecting the homes. In our graph, each node was a home. The length/weight of each edge was defined by the street distance between the two homes it connected. In line with our belief that information sharing should only occur between homes in the same immediate neighborhood, we only connected homes that were within 0.5 km of one another.

With the graph fully constructed, we could finally conduct diffusion. The diffusion process is visualized in the figure below. In diffusion, the values in a graph are smoothed between nodes. In our case, the lead probabilities are smoothed between connected homes. For example, a high probability lead home located near many low magnitude probability homes will diffuse its prediction strength to its neighbors – its probability of lead will decrease. In the case of 1925 Becker St.

(shown in the figure), we see a low probability lead home located near many high probability lead homes. It borrows prediction strength from its neighbors, and its probability of lead rises. In this way, diffusion allows our model to "correct for" discrepancies between neighbors.



As a result of using diffusion, 1925 Becker St. advanced 502 places in the dig queue. To put this change in perspective, digging 500 homes costs at least $1.5 million. A city with a limited budget may not have the funds to dig an extra 500 homes. So, 1925 Becker St.'s climb in the dig order could represent the difference between replacing and failing to its dangerous pipes.

Future Work

There are numerous future directions for this project. From a modeling standpoint, two there are two avenues which have been discussed and would likely be fruitful.

- *Combine L1 Regularization and Diffusion:* Currently, diffusion operates as a layer regularizing the baseline model predictions. However, we believe that one potential research direction stemming from this would seed the diffusion models with the L1-Regularized model to determine additional gains from diffusion.
- *Run diffusion without baseline probabilities*: As an entirely separate model, diffusion could be constructed to run separately from the baseline model. By beginning with fixed training nodes, information could be diffused across the network of the city. This may yield a meaningfully different representation of the lead density in Flint (or other cities) and could bring about further understanding.

In terms of additional directions, there are two more which come to mind:

- *Ecological Validity*: Repeat this analysis for alternative cities / datasets to better understand whether the effects from Flint are unique, or whether this information and modeling technique can be extrapolated to other geographies as well.
- *Optimization with Hit Rate*: Our evaluation algorithm presents a very basic, greedy approach to presenting model performance. It is possible that some other dynamic optimization method may produce superior results, either for diffusion or the baseline model.

Appendix A: Additional Hit Rates

Figure A1: Comparison of the Hit Rate Curve by Parcel Investigated and Prediction Probability

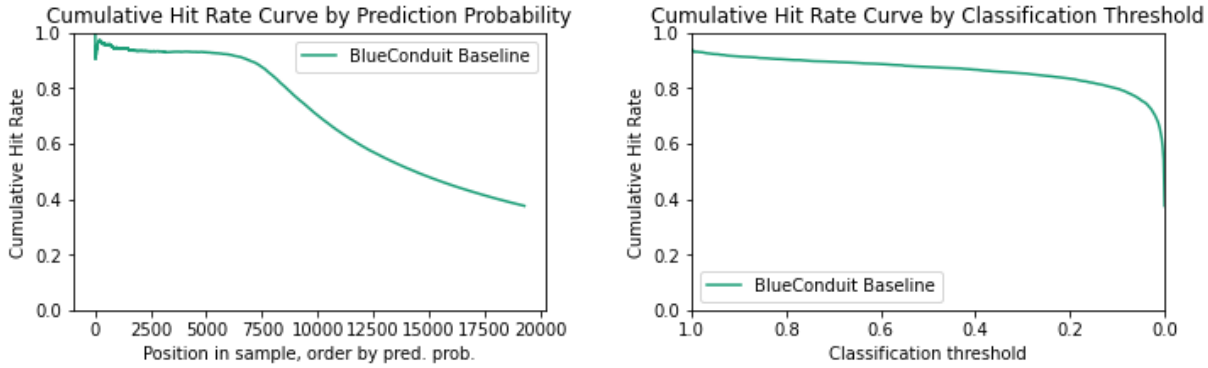
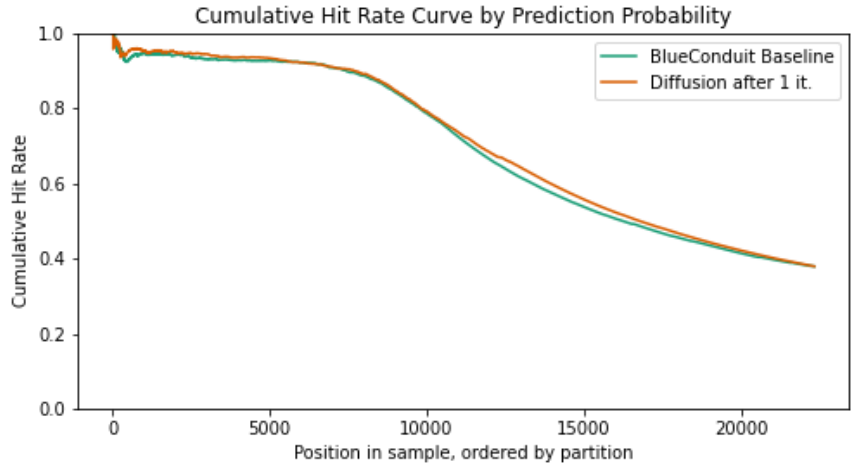
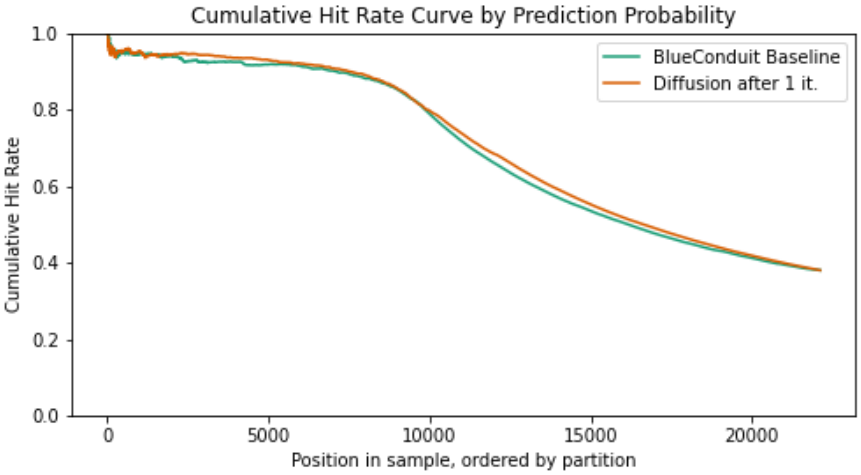
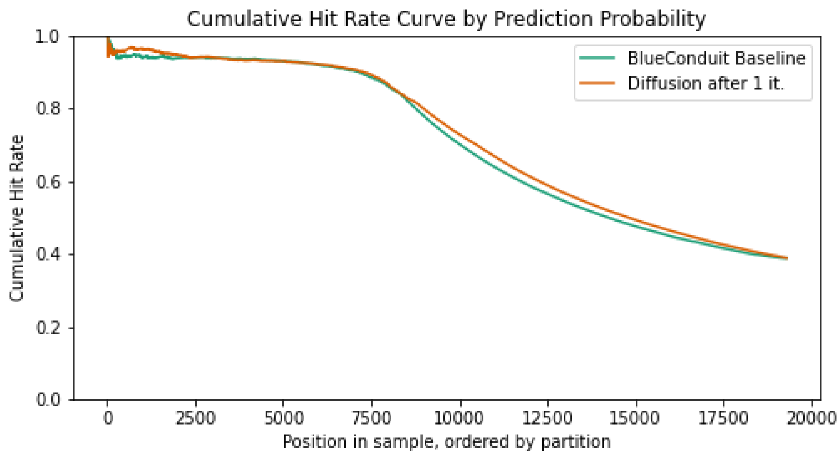


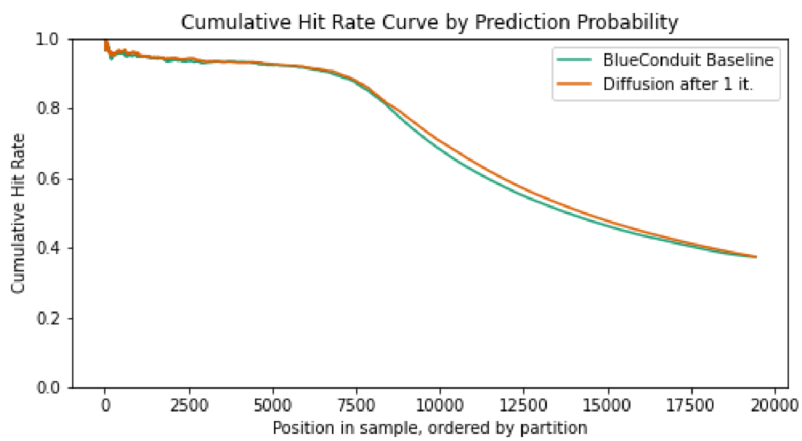
Figure A2: Diffusion Hit Rate Curves between Splits
22 x 22 Hexagon Tiling, Splits #2 & 3

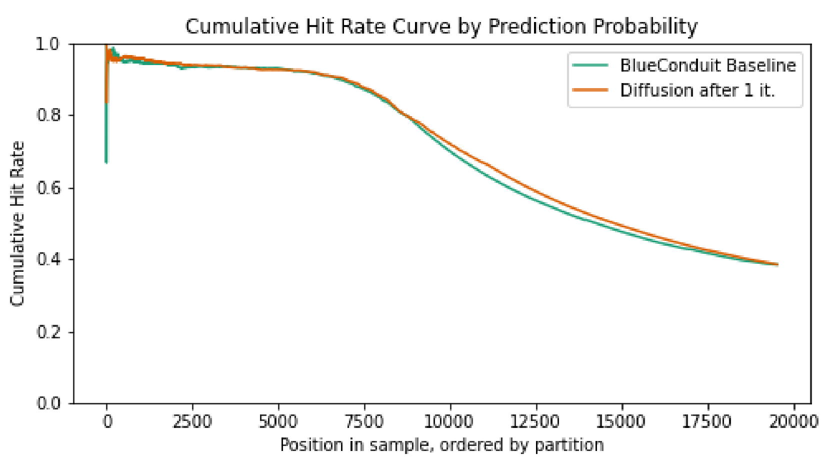




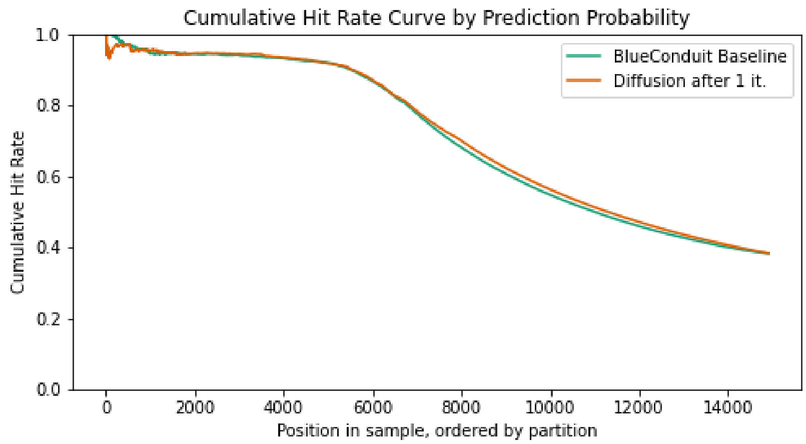
47 x 47 Hexagon Tiling, Splits 1-3

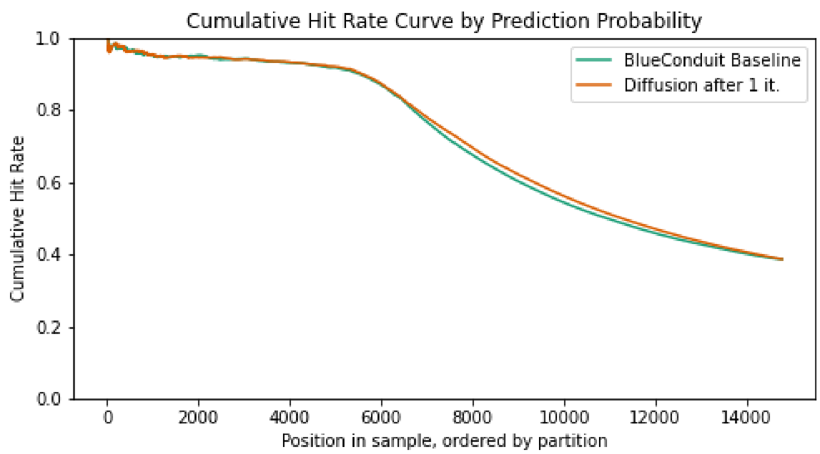


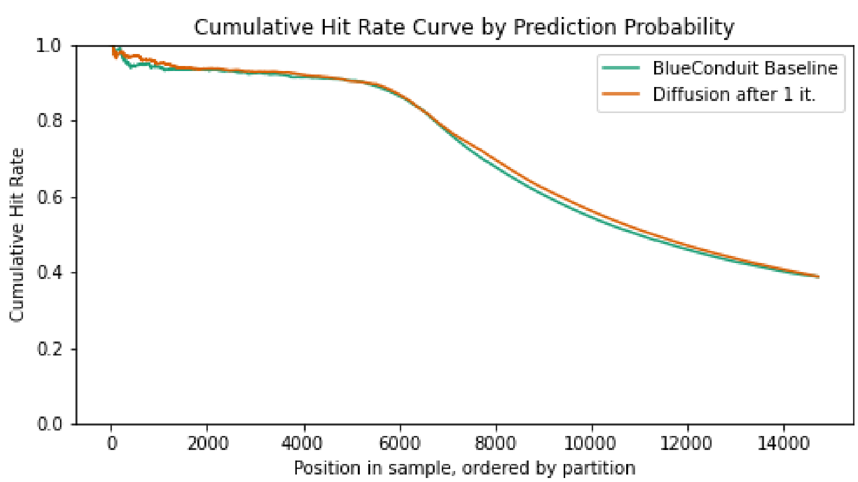




99 x 99 Hexagon Tiling, Splits 1-3

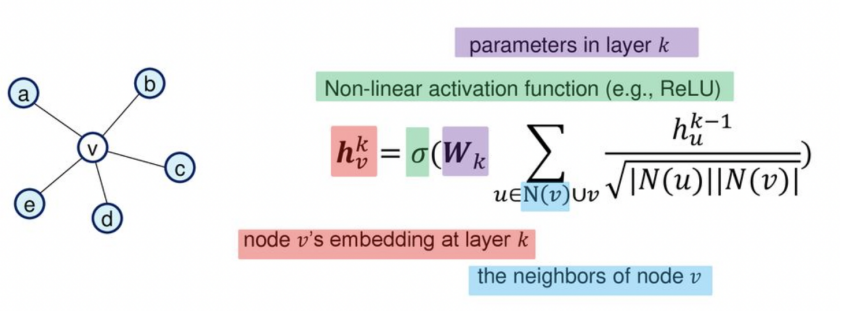






Appendix B: Graph Neural Network models attempted

Graph Convolutional Network (GCN or Conv-GNN)



http://keg.cs.tsinghua.edu.cn/jietang/

Implemented directly using the official Keras tutorial in order to research best practices (e.g. using batch normalization in all fully-connected MLP components).
Features: XGBoost probabilities, Year Built, Residential Building Value, Land Value, Parcel Acres
Model settings:

- Layers: preprocess, conv1, conv2, conv3, postprocess, dense
- We use a standard setup for a fully connected MLP: two layers of [BatchNormalization, Dropout with 50%, and Dense with 8 hidden units]
- Each convolutional layer uses a *prepare*, *aggregate*, and *update* function. Our prepare function is our standard MLP.
- We use a skip connection between the call of each convolutional layer:

  Output = *Dense* (*Postprocess* (X + *Conv3* (X + *Conv2* (X + *Conv1* (*Preprocess*(X))))))

- Our preprocess & postprocess layers are each our standard MLP
- The final dense layer is linear, outputs logits directly

Graph Attention Network (GAT)

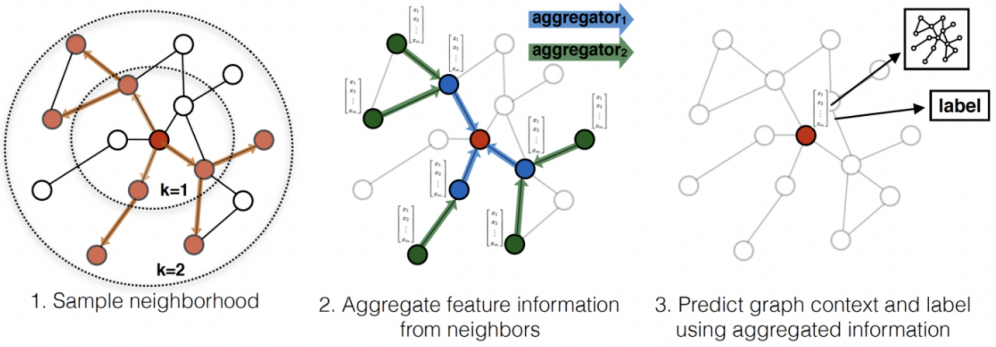
$$h_v^k = \sigma(\sum_{u \in N(v) \cup v} \alpha_{v,u} W^k h_u^{k-1})$$

Learned attention weights

http://keg.cs.tsinghua.edu.cn/jietang/

Implemented using the StellarGraph library: see here for the GAT demo we followed
Features: XGBoost probabilities & ALL features collected by Blue-Conduit
We use multi-headed attention using 8 heads. Each attention layer has 8 hidden units and uses Dropout with 50%.

GraphSAGE: http://snap.stanford.edu/graphsage/

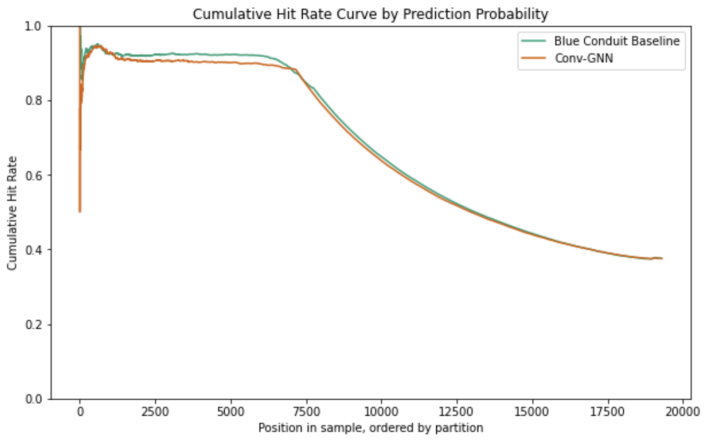


Implemented using the StellarGraph library: see here for the GraphSAGE demo we followed.
Features: XGBoost probabilities & ALL features collected by Blue-Conduit
Optimization notes:

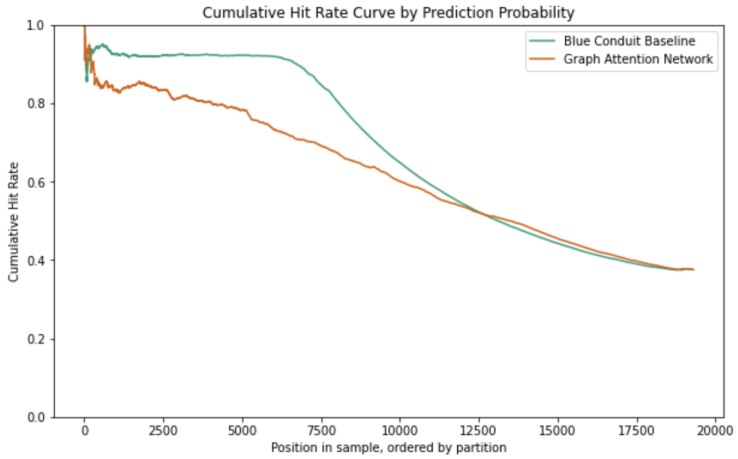
- Batch size: 256
- Neighborhood sampling: K=2 layers of sampling, # neighbors per sample [15, 10]
- Hidden layer embedding sizes [64, 64]

Hit Rate Curves

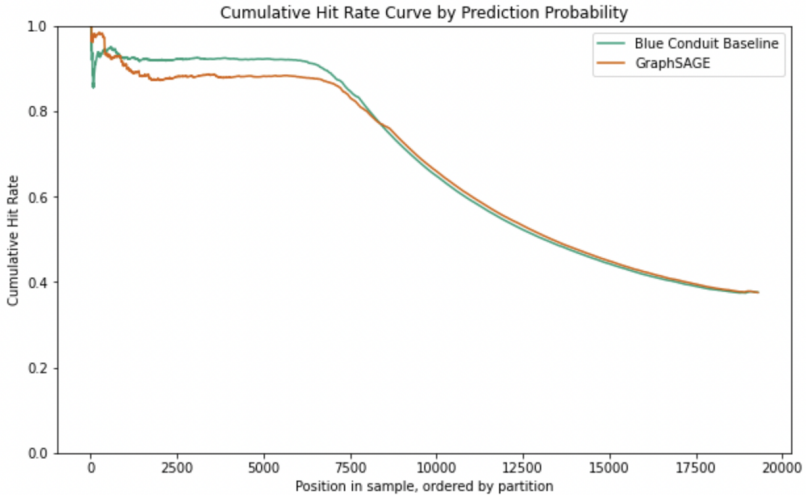
Conv-GNN:



GAT:



GraphSAGE:



Appendix C: Hyperparameter Tuning Results

A. Gaussian Processes

Hyperparameter	**Values tested**	**Notes**
Lengthscales	n/a Learned optimal values: **[Year: 31.16, Lon: 25.64, Lat: 29.27]**	The GPFlow (library we used for GP implementation) Stochastic Variational Gaussian Process model allows for this parameter to be learned directly during

		the optimization
Variance	n/a Learned optimal value: **5.33**	(same as above)

B. Diffusion

To tune the hyperparameters of the network diffusion model, we compared across the parameters described in the table below. All results were compared via grid search. Notably, we compared for the train data size of 10% and across spatial resolutions. That is, we did not treat the size of the area as a hyperparameter, instead looking for robust hyperparameters across configurations. Selected hyperparameters are bolded; full results of hyperparameter tuning is available through our GitHub repository.

Table B1: Hyperparameters considered for diffusion

Hyperparameter	**Values tested**	**Notes**
Update steps	**1**, 2, 7	
Distance type	Haversine distance, **Road distance**	See Section II.B for a discussion of distance metrics
Self-weight	0.5, **0.6**, 0.7, 0.8, 0.9	All update steps are calculated via weighted average where some value λ controlling weight on original value and neighbors.
k-Neighbors	5, 10, **50**, 100	
Weight kernel	**Distance**; Square root of distance	

C. Graph Neural Network

Hyperparameter	**Values tested**	**Notes**
Batch size	64, 128, **256**, 512, 1024	

# layers of sampled neighbors	**2,** 3	
# Neighbors sampled per layer	2 layers: [10, 5], **[15, 10]**, [20, 10], [25, 15], [25, 25], [64, 32] 3 layers: [30, 20, 10]	
Hidden layer sizes	[16, 16], [32, 32], **[64, 64]**, [128, 128]	
Dropout rate	20%, **50%**	
Adam optimizer learning rate	**1e-1,** 1e-2, 1e-3	