

ALF-Score++ - Transferability of a Predictive Network-Based Walkability Scoring System

Ali M. S. Alfosool (✉ alias@mun.ca)

Memorial University of Newfoundland

Yuanzhu Chen

Queen's University

Daniel Fuller

Memorial University of Newfoundland

Article

Keywords: walkability, transfer-learning, ALF-Score++

Posted Date: December 1st, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1079932/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Scientific Reports on August 18th, 2022.
See the published version at <https://doi.org/10.1038/s41598-022-17713-y>.

1 ALF-Score++ — Transferability of a Predictive 2 Network-Based Walkability Scoring System

3 Ali M. S. Alfosool¹, Yuanzhu Chen^{2,*}, and Daniel Fuller¹

4 ¹Memorial University of Newfoundland, Canada

5 ²Queen's University, Canada

6 *Corresponding Author

7 ABSTRACT

Walkability is an important measure with strong ties to our health. However, there are existing gaps in the literature. Our previous work proposed new approaches to address existing limitations. This paper explores new ways of applying transferability using transfer-learning. Road networks, POIs, and road-related characteristics grow/change over time. Moreover, calculating walkability for all locations in all cities is very time-consuming. Transferability enables reuse of already-learned knowledge for continued learning, reduce training time, resource consumption, training labels and improve prediction accuracy. We propose ALF-Score++, that reuses trained models to generate transferable models capable of predicting walkability score for cities not seen in the process. We trained transfer-learned models for St. John's NL and Montréal QC and used them to predict walkability scores for Kingston ON and Vancouver BC. MAE error of 13.87 units (ranging 0-100) was achieved for transfer-learning using MLP and 4.56 units for direct-training (random forest) on personalized clusters.

9 Introduction

10 Walkability is a concept that many researchers have used to operationalize characteristics of the environment that support
11 walking. Although there are multiple conceptual definitions of walkability in the literature ¹⁻⁷ there is no single agreed-upon
12 operational definition of walkability. There are a number of existing walkability measures that provide walkability scores
13 for Canada each with different strengths and limitations. While a number of city-specific walkability measures have been
14 developed, there are two prominent, national level walkability measures available in Canada: Walk Score and the Canadian
15 Active Living Environments measure (Can-ALE). These measures each has different strengths and limitations. Both Walk Score
16 and Can-ALE are heavily used/cited ⁸⁻¹¹. But there are some noticeable drawbacks and opportunities to improve these measures.
17 These limitations are important and are the result of limited interdisciplinary work between the fields of computer science,
18 public health, and urban planning. The important limitations of previous works include: incomplete use of road structures, lack
19 of predictive models, low spatial resolution, lack of user opinion, lack of personalization, and limited transferability to new
20 cities. For the purpose of comparison in this paper, we choose Can-ALE¹² as it is commonly used by researchers and end-users
21 alike.

22 In our previous paper, we created the Active Living Feature Score - ALF-Score ¹³, a completely new approach to measure
23 walkability. This predictive approach allows us to use various important features currently not utilized by most existing
24 walkability measures, such as road network structure as nodes, road embedding, complex centrality measures, and user opinion
25 along with a new approach of using machine learning to estimate walkability scores. By using a predictive approach ALF-Score
26 is able to generate walkability scores with high spatial resolution allowing us to predict walkability scores for any point along a
27 road network. Moreover, in our paper ALF-Score+ ¹⁴ which followed after ALF-Score, we showed an extension of ALF-Score
28 which utilizes user and system defined user-demographics to create individual sociodemographic profiles to develop profile
29 clusters. User labels and profile clusters are then used by ALF-Score's pipeline to generate machine learning predictive models
30 capable of estimating personalized walkability scores specific to each profile cluster. Examples of cluster profiles include for
31 example groups of volunteer participants with similar or varying demographics who may have a similar view of walkability.
32 For example, female professionals in their 20's and 30's who do not live alone and have no children who perceive walkable
33 distances as being greater than 1500 meters, may perceive walkability similarly and can form a profile cluster.

34 In this paper, we introduce ALF-Score++ which is another extension of ALF-Score. ALF-Score++ focuses on transferability.
35 The overall goal of ALF-Score++ is to ensure the pipeline is capable of generating reproducible predictive walkability models
36 that are transferable and able to generate walkability scores for new cities without the need for any new user data (zero-user-input
37 approach, further explained in methods section) or training.

38 Our main objective is to ensure our pipeline can generate transferable models. Transfer learning is yet another missing

39 technique from many of the existing walkability measures. Being able to generate reproducible and transferable predictive
 40 walkability models is an important component of which ALF-Score++ takes advantage of in two ways: 1) by gaining the ability
 41 to utilize previously learned knowledge when directly generating walkability scores for new cities (zero-user-input), 2) by using
 42 this previously learned knowledge as a base to train new models which can lead to reduced training time, improved accuracy,
 43 reduced resource consumption, and reduction in the labels required for supervised learning tasks. A well generalized model will
 44 have the capability of transferring its knowledge to various cities never seen during its training to generate accurate walkability
 45 scores in a fraction of the time without the need for any new user input within the target city.

46 In this paper we will highlight application of ALF-Score++ to three new cities of Kingston Ontario (ON), Vancouver British
 47 Columbia (BC), and Montréal Quebec (QC).

48 Background

49 Ensuring ALF-Score’s pipeline does not engage in repeated wasteful activities is one of the sub-objectives of this research. This
 50 is particularly important since road networks can vary in size with some cities being very small (eg. with a population of a few
 51 hundred) while some other cities could be very large and dense (eg. Tokyo, Japan with a population of over 37 million people
 52 in just one city). Table 1 shows a list of various cities alongside their network size, number of POIs, population and total land
 53 area size. Processing data from St. John’s, NL as opposed to data from Toronto, ON will have significantly different resource
 54 requirement and time consumption due to the change in the the size of the city leading to an extended set of complexities
 55 introduced into the network. If the algorithms are not optimized, this difference in requirements may lead to infeasibility of the
 56 research. In this research we have experimented with all cities mentioned in Table 1; however, we will only highlight the results
 57 for Kingston ON, Vancouver BC, and Montréal QC.

Table 1. List of road networks for various cities with their network and POI sizes that have been experimented with in this research. For brevity, in this paper we mostly focus on 3 cities of Kingston, Vancouver and Montréal. Nodes and edges are extracted from road networks. Population density and the total land area information are excerpted from Wikipedia.

<i>City</i>	<i># of Nodes</i>	<i># of Edges</i>	<i># of POIs</i>	<i>Population</i>	<i>Total Land Area</i>
Victoria, BC	6,770	8,593	3,318	85,792	19.47 km ²
Kingston Metro, ON	3,427	4,769	813	161,175	1,906.82 km ²
St. John’s Metro, NL	5,364	6,851	592	205,955	804.63 km ²
Vancouver Metro, BC	45,125	60,299	13,321	2,463,431	2,878.52 km ²
Montréal Metro, QC	76,663	114,414	10,045	4,247,000	4,604.26 km ²
Toronto Metro, ON	479,520	Over a million	23,930	6,417,516	5,905.71 km ²

58 Transfer learning is the process of re-utilizing the knowledge learned from a task in other tasks. In many machine learning
 59 approaches, solving a single task at hand has been the main focus, but now development of approaches that help with transfer
 60 learning has become a very popular focus in the recent years¹⁵. As with most real-world problems, specifically in machine
 61 learning, collecting labelled data is a time consuming, expensive¹⁶ and difficult task. Transfer learning uses the knowledge
 62 learned from previous problems to solve new but related problems¹⁷. As a result of its approach, transfer learning can help
 63 reduce training time, resources and the required labeled data,¹⁸ as well as improve overall accuracy. Weiss et al.¹⁹ provide a
 64 much more formal definition of transfer learning as the following: “given a source domain \mathcal{D}_S with a corresponding source task
 65 \mathcal{T}_S and a target domain \mathcal{D}_T with a corresponding task \mathcal{T}_T , transfer learning is the process of improving the target predictive
 66 function $f_T(\cdot)$ by using the related information from \mathcal{D}_S and \mathcal{T}_S , where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ ”.

67 The general idea behind transfer learning is to apply a model that was previously trained on labelled data (in case of
 68 supervised learning) to another similar task with little data available and instead of starting from scratch, start with some
 69 existing knowledge and captured patterns. Transfer learning is typically used in computer vision, for example, the weights of a
 70 model that was trained to detect apples could be transferred for another task of detecting fruits. In this case, instead of training
 71 the new model to detect apples from scratch, the knowledge about detecting apples are transferred and the algorithm now looks
 72 to learn how to detect other fruits. Transfer learning is a technique that not only requires significantly less data for training, but
 73 it will also speed up the training process²⁰.

74 Transfer learning falls under representation learning with the goal of using the same representation in various tasks.
 75 According to Ian Goodfellow²¹ transfer learning can be viewed as a particular form of multi-task learning where it normally
 76 revolves around supervised learning. Although, transfer learning can also be used to solved unsupervised learning tasks. The
 77 goal of transfer learning is to take advantage of previously trained models and to extract knowledge that would be useful
 78 in the new task. However, transfer learning is also very useful to directly generate predictions in another environment and

79 for other tasks without any more learning ²¹. There are a few approaches to transfer learning including feature extraction,
80 training generalized models, and use of existing pre-train models. When it comes to feature extraction, determining the best
81 representation for the problem at hand is a key task which if done properly, can often lead to much better and more accurate
82 results. Carefully selected features can often lead to a powerful and well-generalized model that can be applied to various
83 related problems. Furthermore, using already available pre-trained models is yet another, very popular option. In fact there are
84 numerous pre-trained models available online that provide ready weights for many popular tasks such as classifying certain
85 types of images, object detection and object tracking. It is important to highlight that this approach only requires access to
86 a previously trained model, and not the entire dataset. Additionally, another approach to solve a task using transfer learning
87 where there is not enough data available and no pre-trained models can be found, is to take the previous approach a step further
88 and train models that are designed for another but similar task and that have an abundance of data. These models can then act
89 as a starting point to address the original task. To highlight the difference with the previous approach, in this technique to solve
90 task A, we will be doing our own training on a similar task B. Once we are satisfied with the model, we can now transfer and
91 reuse this knowledge. Goodfellow in his book ²¹ further discussed two extreme forms of transfer learning, namely: 1) one-shot
92 learning - which only one labeled example of the transfer task is given while, 2) zero-shot learning, has no labeled example
93 given.

94 As an overview to the data used in this research, the general structure of our road network and feature set remains the
95 same to one described in our paper ALF-Score ¹³. We collected a small set of user opinion data containing 1,050 user entries
96 covering 895 unique locations for the city of St. John’s, NL. This includes $n = 40$ users with $n = 20$ (50 %) women with an
97 average age of 48.6 (standard deviation = 17.1). The most commonly reported walkable distance was 800-1000 meters while
98 ten participants (25 %) reported living alone whereas 14 participants (35 %) reported living with children with the average
99 number of children being 2.6 (standard deviation = 1.2). The most commonly reported professions were Retired $n = 8$ (20 %),
100 Professor $n = 4$ (10 %), and Nurse $n = 4$ (10 %). In addition, more user opinions have been collected which are specific to the
101 city of Montréal, QC containing 785 user entries covering 775 unique locations. Similarly, this includes $n = 21$ users with
102 $n = 13$ (62 %) men with an average age of 40.95 (standard deviation = 17.29). The most commonly reported walkable distance
103 was 1200-1400 meters while five participants (24 %) reported living alone whereas 8 participants (38 %) reported living with
104 children. The most commonly reported professions were Professional $n = 4$ (19 %), Professor $n = 3$ (14 %), and Retired $n = 3$
105 (14 %).

106 Results

107 In this research, we were able to successfully achieve transferability for ALF-Score++. First, using the newly collected user
108 opinion data for the city of Montréal QC, we were able to achieve a consistency of 99.6% during the GLEPO processing stage.
109 While various feature combinations and machine learning techniques were experimented with, we were able to achieve our
110 lowest prediction MAE error (Matching approach) using random forest shallow model at 11.87 units (Figure 4 top left) while
111 MLP was the best performing deep model with an MAE error of 13.87 units. Table 2 highlights some of the techniques and
112 feature combinations used to generate ALF-Score for the city of Montréal using the user opinion collected from the same city.

Table 2. Exploration of various machine learning techniques and feature combinations over an 80-20 data split (Matching approach) for the city of Montréal, QC reflecting their top performing accuracy. Results represent MAE error over a range of 0-100 units.

Technique	<i>POI</i>	<i>POI + Network</i>	<i>POI + Embedding</i>	<i>Network + Embedding</i>	<i>All</i>
<i>Random Forest</i>	19.65	18.20	17.13	15.47	11.87
<i>MLP</i>	26.65	24.08	23.44	23.56	21.91
<i>SVM</i>	29.03	31.04	29.78	23.63	21.74
<i>Decision Tree</i>	21.65	31.87	34.23	24.45	21.49

113 As we explored in the background section, the goal of transfer learning is to take advantage of previously trained models,
114 for instance models trained on the city of St. John’s NL in our previous works, to essentially extract knowledge that could
115 be useful when applied to training new models (Combined approach) for new cities. However, transfer learning is also very
116 useful to directly generate predictions for new cities without any more learning (Zero-user-input approach). Zero-user-input
117 was our first transfer learning approach. We used our best model trained on data for the city of St. John’s using random forest
118 to predict ALF-Score walkability for the city of Montréal (Figure 4 top right). This resulted in a correlation of 0.4 compared
119 to the predictions generated by a model that was trained purely on Montréal’s user data (Figure 4 top left). Furthermore, our
120 second approach of using previously trained models (MLP) towards training new MLP models (Figure 4 bottom left) led a
121 much higher correlation of 0.79 compared to the model only using the data from one city. We believe this promising model well

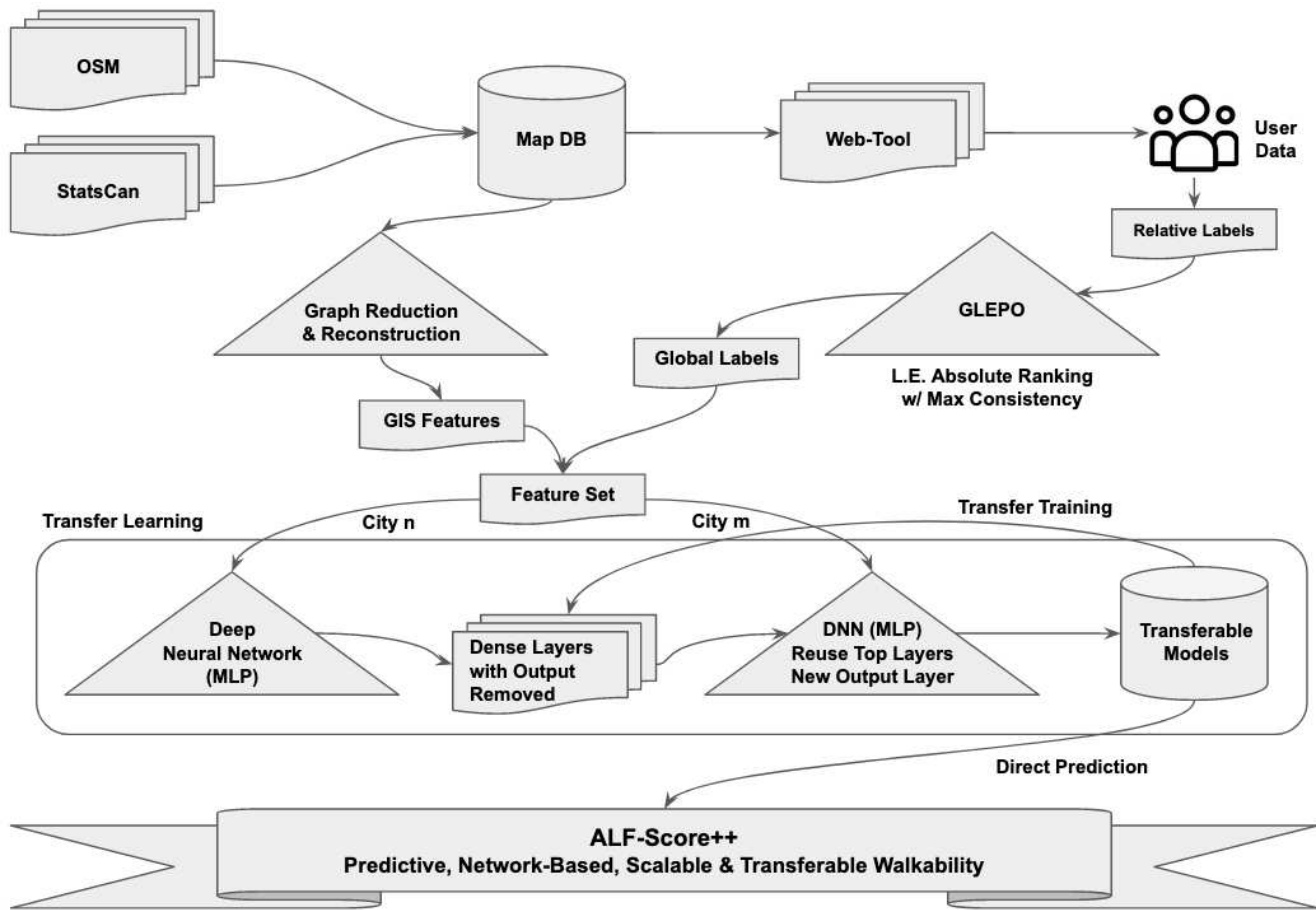


Figure 1. ALF-Score++ utilizes features similar to that of ALF-Score and ALF-Score+ such as road network structure, POI, centrality measures and road embedding. GLEPO’s linear extension of user opinions that produces a global view of relative user opinions is then aligned with the features as an input to the machine learning processes. Models trained by ALF-Score++ are applicable to cities previously seen and unseen by the algorithms during the training processes. Walkability estimates that are produced through trained models will have a high spatial resolution, be representative of user opinion and provide a better insight of different regions and neighbourhoods. (Figure drawn by the authors.)

122 utilizes the transferred learning in conjunction with the new learning gained by training over new data to identify additional
 123 patterns that may have not been fully captured by a model trained on a small set of user data from a single city.

124 We can observe, that among the top 150 features (out of 668 features), 128 of them belong to the road embedding feature
 125 list and accounts for all road embedding features. This is highlighting the importance of road embedding with regards to
 126 predicting walkability score based on user submitted ground-truth. Additionally, among top 150 features, only 14 belongs to
 127 POIs which contributes to 530 features. Furthermore, among the top 150 features, 6 belongs to centrality features out of the
 128 total 10 centrality features.

129 The road embedding features account for 0.778486799 importance over 128 features while representing only 19% of the
 130 overall features. The centrality features account for 0.039919843 importance over 10 features, and the POI features account for
 131 0.169245465 importance over 530 features while representing over 79% of the features.

132 Eccentricity accounts for the highest centrality importance among the 10 features; however, it is contributing almost 33% to
 133 the overall centrality importance which is rather a important amount when considering there are 9 other centrality features as
 134 well. Highest ranked POI is restaurants within 600 meters which contributes to almost 9% of all POI importance among 529
 135 other POI features. Furthermore, it is very interesting to see 8 out of the top 10 POIs are either restaurants or cafes, while bars
 136 within 1,800 meters and benches within 1,800 meters amount to the remaining top 2 POIs . This points to the possibility of
 137 many people seeking out to find places to socialize, with light entertainment and possibility to gather with friends and family.
 138 Especially, since the user data in this research was collected post COVID-19 pandemic, this may show an underlying effect of
 139 the pandemic’s isolation as to change people’s priority/perception to place an important value on socializing.

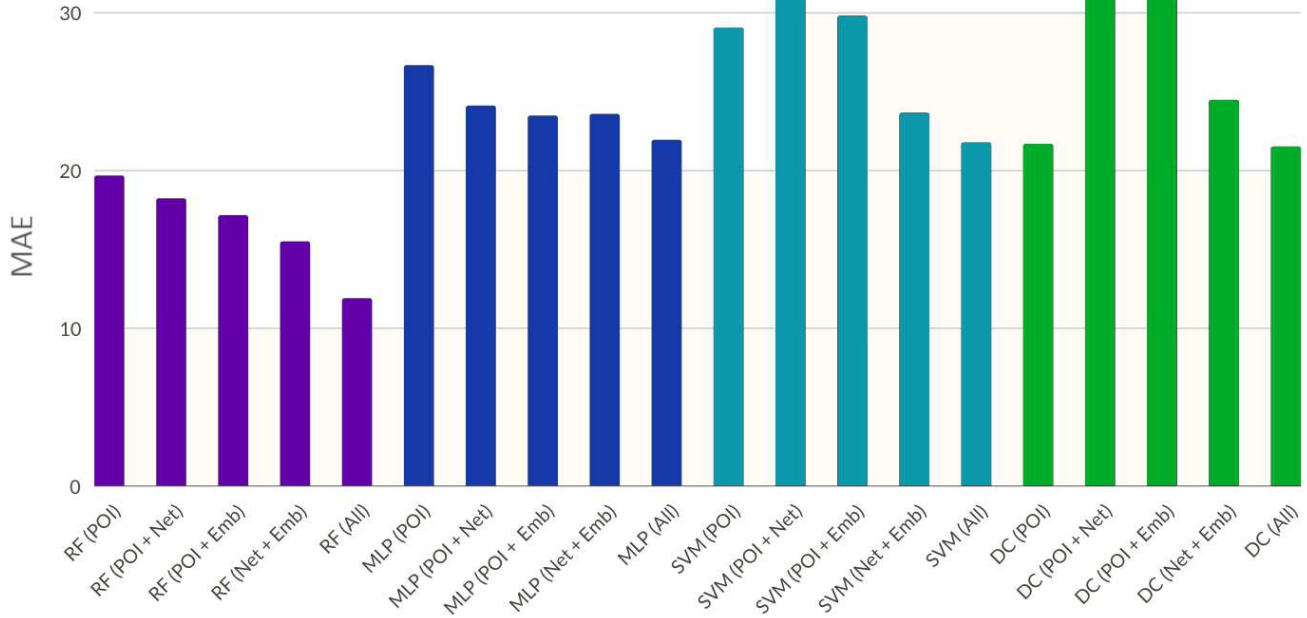


Figure 2. Experimentation results of four machine learning techniques over five feature combinations for the city of Montréal, QC with a data split of 80-20. The bars represent MAE error over a range of 0-100 units. RF: Random forest, MLP: Multi Layer Perceptrons, SVM: Support Vector Machine, DC: Decision Tree. RF provides the best performance overall. (Bar plot generated through matplotlib²² Version 3.4.3 from matplotlib.org.)

140 The next step is to utilize the zero-user-input approach of the transfer learned model trained on the user data collected from
 141 the two cities of St. John’s NL and Montréal QC which have different structures, and applying this model directly to a third and
 142 a fourth cities of Kingston ON and Vancouver BC, which the model has never seen before, and generate ALF-Score walkability.
 143 In Figure 7 we can see the ALF-Score walkability (right) compared to Can-ALE scores (left) for the city of Kingston, ON. At
 144 the first glance we can easily observe the variation in spatial resolution between these two methods with ALF-Score capturing
 145 the walkability of the region in a much greater depth. While Can-ALE shows some variation among different dissemination
 146 areas (DA), only the city center is highlighted with visible green and marked as walkable. Although ALF-Score++ agrees with
 147 Can-ALE with assigning higher walkability scores to the city center, the first major differentiator among the two is that in
 148 Can-ALE’s higher walkability is given to the central and highly populated areas of the city center whereas in ALF-Score++,
 149 while central region is ranked with higher walkability, ALF-Score++ recognizes the core as slightly less walkable compared to
 150 locations surrounding the core of the city center. Specifically, ALF-Score++ favours waterfront walkways and paths as more
 151 walkable as opposed to Can-ALE. For instance, the area near to Leon’s Centre on Ontario Street is known to be a walkable area
 152 and is ranked with high walkability through ALF-Score’s zero-user-input approach whereas it is ranked with a significantly
 153 lower walkability score by Can-ALE.

154 Additionally, ALF-Score captured a cluster of greener/more walkable spots close to students housing and living quarters
 155 near Queen’s University. While this area is popular among many students, faculty and other members of the public, Can-ALE
 156 was unable to capture it due to its area-based structure and lower spatial resolution. Moreover, we observed various other
 157 areas that ALF-Score++ ranked as walkable whereas Can-ALE failed to capture their actual walkability due to its lower
 158 resolution and granularity. For instance, the Division St. — Dalton Ave. — Benson St. region (which falls under multiple
 159 DAs) is ranked with low walkability scores by Can-ALE whereas ALF-Score captured and distributed much more refined and
 160 relatable walkability scores to varying spots where there are many restaurants, stores and other popular places. Furthermore,
 161 the walkability of Point Frederick Peninsula (across the LaSalle Causeway bridge) is in the red zone of the Can-ALE scores
 162 while ALF-Score suggests the opposite for the region. This region houses multiple military campuses with varying facilities
 163 and is deemed walkable.

164 Figure 8 shows the ALF-Score++ walkability (right) compared to Can-ALE scores (left) for the city of Vancouver, BC. The
 165 ALF-Score++ for this region is generated based on a zero-user-input approach and similar to ALF-Score++ for Kingston, we

Table 3. Exploration of the three experimentation approaches (1) Matching, (2) Combined and (3) Zero-user-input over 5 different feature combinations and 2 different data split approaches based on data from the cities of St. John’s NL and Montréal QC. Results represent MAE error over a range of 0-100 units.

MLP	POI	POI + Network	POI + Embedding	Network + Embedding	All
<i>St. John’s (STJ on STJ (100%))⁽¹⁾</i>	27.55	26.22	22.23	21.91	17.88
<i>Montréal (MTL on MTL (100%))⁽¹⁾</i>	26.65	24.08	23.44	23.56	21.91
<i>STJ on MTL (100%)⁽³⁾</i>	n/a	n/a	n/a	n/a	32.44
<i>STJ on STJ (50%) + MTL (100%)⁽²⁾</i>	26.87	25.10	23.55	19.31	15.77
<i>STJ on STJ + MTL⁽²⁾ (rand 80-20)</i>	25.87	23.74	21.45	20.23	14.12
<i>MTL on STJ (100%)⁽³⁾</i>	n/a	n/a	n/a	n/a	33.89
<i>MTL on STJ + MTL⁽²⁾ (rand 80-20)</i>	25.11	22.23	21.67	20.11	16.23
<i>MTL on STJ (100%) + MTL (50%)⁽²⁾</i>	27.67	24.86	14.43	21.51	16.73
<i>MTL on STJ (100%) + MTL (80%)⁽²⁾</i>	24.84	20.17	19.92	18.36	13.87
<i>MTL on STJ (100%) + MTL (20%)⁽²⁾</i>	29.66	25.34	25.73	22.89	18.34

Feature	Importance
Eccentricity	0.01316184
Stress	0.004649347
Betweenness Centrality	0.004590322
Average Shortest Path Length	0.0043923
Topological Coefficient	0.003773664
Neighborhood Connectivity	0.003381009
Radiality	0.002233024
Closeness Centrality	0.001581954
Clustering Coefficient	0.001386535
Degree	0.000191574

Table 4. Feature importance for all centrality features (10 features in total) which contribute to 4.1% of the total feature importance.

Feature	Importance
restaurant_600	0.014984423
bar_1800	0.010154083
cafe_1400	0.007755144
cafe_1600	0.007659399
cafe_2000	0.007125045
cafe_1800	0.005620239
restaurant_1000	0.005089702
restaurant_1400	0.004664054
restaurant_1400	0.004664054
bench_1800	0.003845234

Table 5. Feature importance for top 10 (from 530) POI features. The entire 530 features contribute 17.1% to feature importance.

166 can observe high spatial resolution as opposed to Can-ALE’s low spatial resolution for the same area. To look further into this
 167 region, we can start by observing the University of British Columbia campus where Can-ALE highlights the inner campus
 168 area (left side) with light orange while the outer campus area (right side) remains darker orange. ALF-Score++ picks up on
 169 the fact that the right area should be more walkable due to bus stops and various facilities commonly used by students and
 170 staff. Additionally, North Vancouver’s walkability appears not to have been captured by Can-ALE where its walkability for
 171 the region is ranging between dark orange and red. In contrast, ALF-Score appears to better capture various popular areas
 172 in North Vancouver that are walkable. Furthermore, the walkability for the Richmond area is barely captured by Can-ALE
 173 with mostly dark orange and red walkability. ALF-Score++ on the other hand is able to capture various walkable areas in that
 174 region. An interesting observation here is the similarity with zero-user-input walkability data generated for the city of Kingston.
 175 Can-ALE typically marks areas close to water as less walkable whereas ALF-Score++ tends to object. ALF-Score++’s results
 176 are positively associated to our collective knowledge of Vancouver and Kingston. We can observe that ALF-Score++ is utilizing
 177 its transferability capabilities to better understand the city structures and find patterns in various associated data to generate
 178 zero-user-input walkability scores for virtually any location on the map.

179 As observed earlier, the combination of user data from just two cities of St. John’s and Montréal allowed us to generate
 180 accurate walkability scores for cities never seen by our algorithms. It is our observation that transfer learning works well in this
 181 application even with a small set of user data. Additionally, we believe as we accumulate more user data, our algorithms will be
 182 able to better capture various patterns in the data leading to an improved accuracy.

183 In this research, we were also able to show ALF-Score++’s pipeline is scalable as data size increases. The pipeline was
 184 optimized to perform well while processing, training and predicting walkability scores for small and large cities alike. One of
 185 the major enhancements to the pipeline was improving the GLEPO algorithm such that the processing time is reduced. This
 186 reduction process went through multiple stages. In our initial trials every iteration of GLEPO took approximately 17 minutes

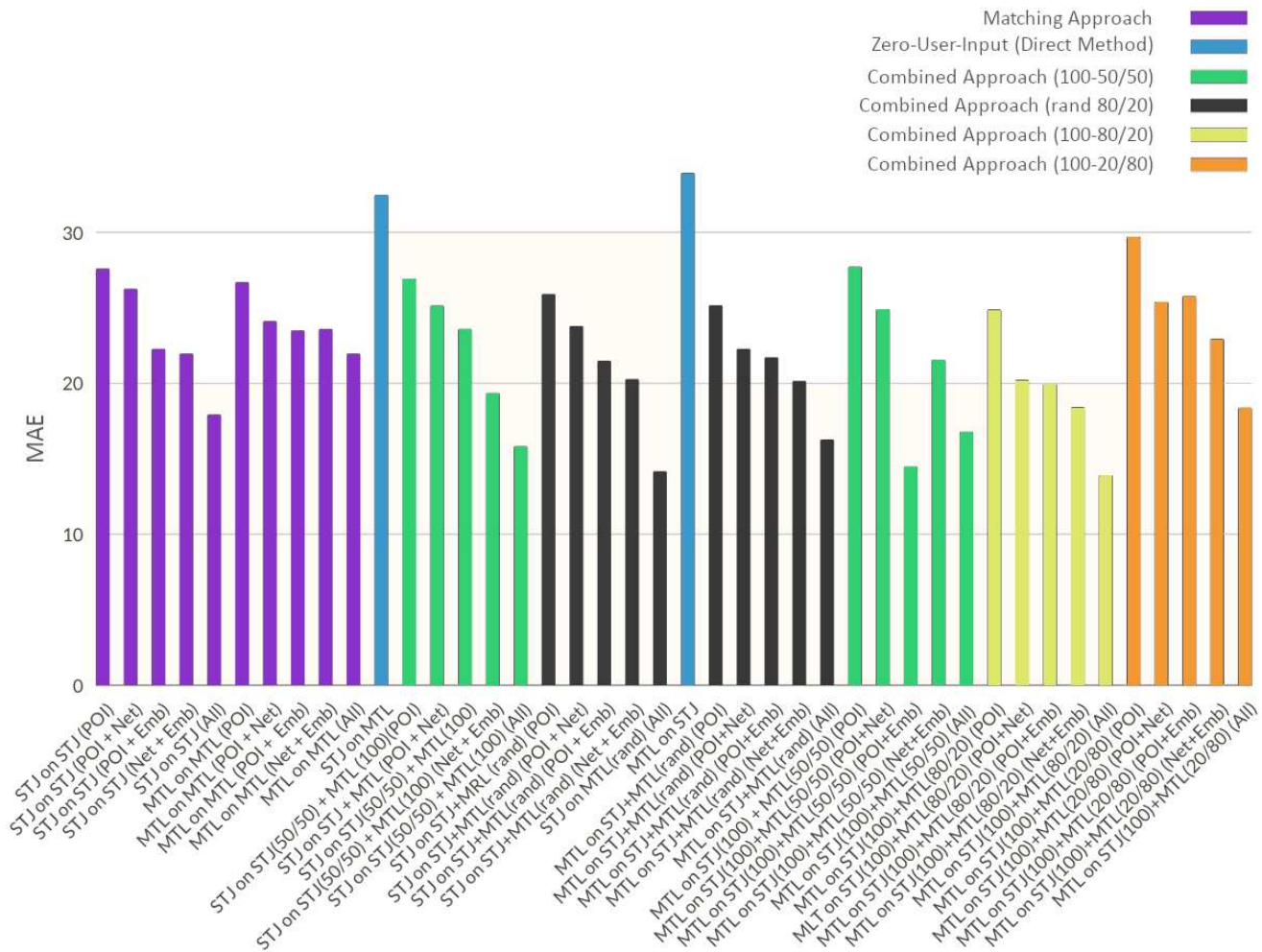


Figure 3. Exploration of 3 approaches (1) Matching, (2) Combined, (3) Zero-user-input. Combined approach is extensively tested with various conditions. One such conditions is different ways of data split to better understand how the data affects the transfer of knowledge in transfer learning while being able to provide a solid training and testing sets. Best performance was observed to be generated through a complete random selection into an 80-20 split. MTL on STJ reflects prediction of scores for Montréal based only on a model trained on St. John’s. MTL on STJ+MTL on the other hand reflects prediction of scores for Montréal based on a transfer learned model on both St. John’s and Montréal. (Bar plot generated through matplotlib²² Version 3.4.3 from matplotlib.org.)

187 on a personal MacBook configured with a 2.2GHz dual-core Intel Core i7 (Turbo Boost up to 3.2GHz) with 4MB shared L3
 188 cache and 8GB of 1600MHz LPDDR3 on-board memory. Over a typical run of the algorithm, we went through approximately
 189 50 iterations totaling to over 14 hours of operation. We found this to be unreasonable. In the final stage of this improvement we
 190 were able to process the same data over the same computer through the newly updated ALF-Score++ pipeline in just under 3
 191 minutes per iteration, a reduction of almost 6 fold. A GLEPO run of 50 iterations will now only take 2.5 hours. Additionally,
 192 after rigorous experimentation and tests, we determined the optimal number of iterations desired for GLEPO algorithm is 50
 193 iterations while the minimum required number of iterations to achieve convergence is 30 iterations leading to a successful
 194 completion of the process within 1.5 hours.

195 Discussion

196 The goal of the overall research is aimed to explore how machine learning can be applied to the spatial domain with application
 197 in public health through generating relevant and meaningful walkability scores with high spatial resolution based on a very
 198 small set of user opinion. In this paper we showed that ALF-Score++’s pipeline is fully capable of scaling up and down to match
 199 the data based on the size of the city and user opinion data and still perform in a reasonably timely manner. Additionally, since

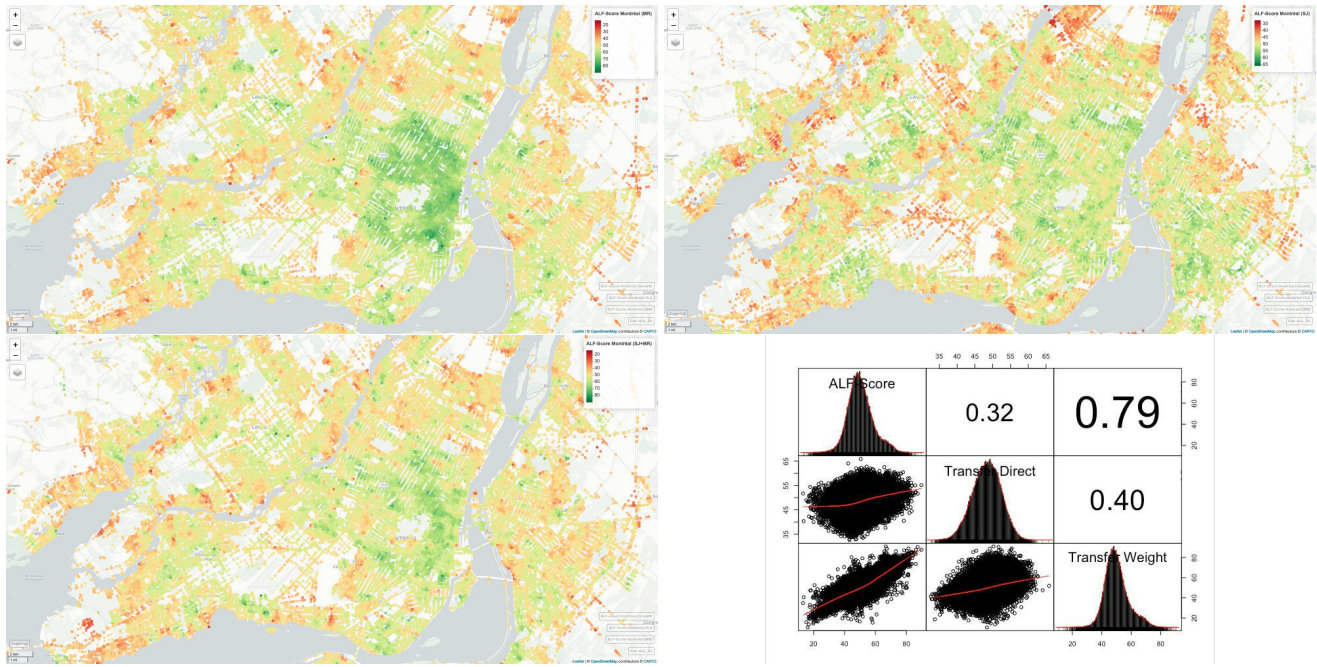


Figure 4. Walkability results produced by 3 separate variations of ALF-Score and ALF-Score++ for the city of Montréal, QC and their correlation. Top left: predictions based on a model only trained for Montréal’s user data. Top right: predictions based on a transferred model only trained for a single city’s user data (St. John’s). Bottom left: predictions based on a model trained for Montréal’s user data while having the previously trained weights for St. John’s user data transferred in its transfer learned training process. Bottom right: correlation between the three variations. The road network for Montréal maintains over 76 thousand nodes. ALF-Score’s walkability scores range between 0-100 units. This range can be adjusted if needed. (Maps generated through RStudio²³ Version 1.2 using mapview package from rstudio.com. Correlation figure generated through RStudio²³ Version 1.2 using PerformanceAnalytics package from rstudio.com)

200 the computational complexity of the pipeline is $O(n^2)$, we expect processing larger cities will perform reasonably and within
 201 the expected parameters. We were able to show that ALF-Score++ can process and generate models for the city of Montréal QC
 202 which is almost 16 times larger than St. John’s NL within a timely fashion without requiring any extended resources while these
 203 models are capable of producing walkability scores with high spatial resolution compared to that of Can-ALE. Figure 9 shows
 204 a comparison between ALF-Score++ walkability scores and Can-ALE walkability scores for four different cities in Canada.

205 Moreover, we saw the power of transferability giving us the upper hand to transfer the knowledge learned from small
 206 cities to predict accurate walkability scores for much larger cities. This leads to many advantages such as reduced resource
 207 requirement and reduced processing time while increasing the flexibility of applicability of our trained models. Furthermore, its
 208 application of zero-user-input transfer learning proved to be a huge success in predicting walkability scores for cities never
 209 seen by the algorithm before and without any prior knowledge about them while utilizing previously learned information and
 210 patterns. Of note, the transfer learning was able to capture both the relative and absolute differences between cities in terms
 211 of walkability. For example, the range of walkability scores assigned to Kingston based on the transfer learning was 30-70,
 212 whereas Vancouver scores based on transfer learning ranged from 30-80, when St. John’s and Montréal were used as training
 213 cities. Developing measures that capture both relative and absolute differences in cities has been an on going challenge, that
 214 transfer learning may be able to solve.

215 We also saw how adding a small set of user opinion from a different region can lead to a much higher pattern recognition
 216 by the models while allowing a better generalization of these models. This generalization can therefore help capture various
 217 common patterns found in different cities without any actual prior knowledge about them.

218 Similar to many machine learning tasks, we believe ALF-Score++’s pipeline can benefit from and train more accurate
 219 models with more data. For instance, collecting small user data samples across various cities and towns could cover a much
 220 more diverse set of user demographic, user opinion, patterns, city and road structures leading to a well-generalized model
 221 applicable to virtually any location on the map. Given enough user information from a few select key cities in Canada (cities
 222 with varying structure and sizes), the model generated through transfer learning of this data will be able to provide accurate
 223 scores anywhere in Canada without the need to train models on data for every individual city leading to a global model. To

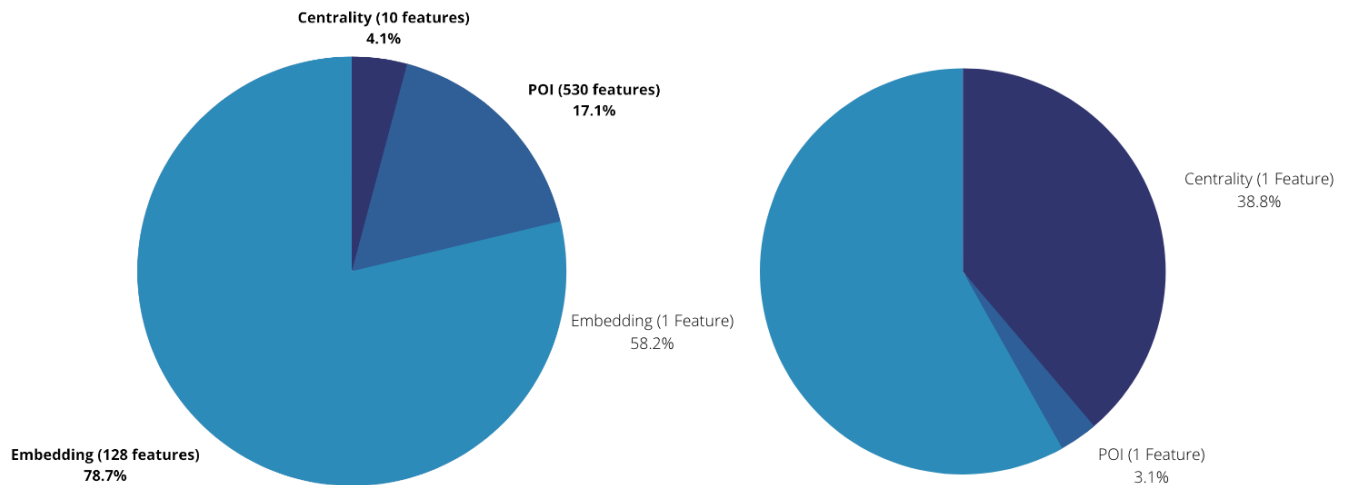


Figure 5. Total contribution to feature importance among 668 features is divided into three categories: 1) centrality, 2) POI, 3) road embedding. Left: Road embedding, while contributing to only 19% of the total features, accounts for 78.7% of the total feature importance, while centrality features contribute to 4.1% and POI features to 17.1% of the total feature importance. Right: when normalized to individual feature importance, we can observe, the highest contribution is by embedding features where each feature contributes to 58.2% of the total embedding contribution of 78.7% where each centrality feature contributes to 38.8% of the total centrality feature importance of 4.1 while each POI feature contributes to only 3.1 % of the total contributing feature importance of 17.1%. (Pie chart generated through matplotlib²² Version 3.4.3 from matplotlib.org.)

224 generate new models to add new data, one does not need to rerun the entire process on the entire data sets. All that is required is
 225 to transfer the knowledge from previously trained models (which can be transfer learned models themselves) and only run
 226 a smaller transfer learning task on the newly collected data. We also believe ALF-Score++ pipeline can be adjusted to be
 227 fully capable of continuous learning. This could be particularly important as changes to road networks are detected. Our
 228 network-based approach combined with continuous transfer learning can help our models detect patterns associated with
 229 various regions, types of road and user demographics and provide accurate predictions for new roads and structures never seen
 230 by the model.

231 As we went through the predictive process, a variation was observed between the performance of shallow and deep models.
 232 Throughout ALF-Score, random forest (a shallow model) was the preferred technique since 1) it performed best across all other
 233 techniques (shallow and deep) achieving an MAE error as low as 4.56 units, 2) its simplicity and powerful approach, 3) faster
 234 processing and prediction compared to MLP. Although, MLP (a deep model) is the main technique used in ALF-Score++ since
 235 deep models are preferred when it comes to transfer learning due to their layered structure. The lowest error was achieved using
 236 MLP at 13.87 units.

237 A side-effect of transfer learning is its generalization. Each city will have its own range of walkability. Small cities may
 238 have a smaller range of walkability whereas larger cities may have a wider range of walkability. When models trained on
 239 small and large cities are combined through transfer learning, the newly trained model will be more generalized. Although this
 240 generalization is very important to be able to take a zero-user-input approach to generate walkability scores for cities never
 241 seen by the algorithms, one must keep in mind that a balance of data must be maintained. As observed earlier (in Montréal's
 242 results), applying a model trained only on a small city might not capture the varying patterns of a larger city and vice versa. It is
 243 important to ensure the transfer learning process maintains a good balance of user data for training, such as using user data for a
 244 small and a large city to build the base model with transferability capabilities. These small samples can prove to be invaluable in
 245 improving the overall quality and accuracy of pattern detection and prediction. Moreover, to further address the generalization
 246 phenomenon happening during the transfer learning phase, we can utilize ALF-Score's personalization extension (ALF-Score+)
 247 to create personalized and transferable models that are generalized to various city structure patterns, yet personalized to specific
 248 individual profile clusters. As demonstrated in our previous work, ALF-Score+¹⁴, concentrating on specific profile clusters
 249 which contain users with similar opinion towards walkability ranking, will significantly improve the overall accuracy of each
 250 personalized model.

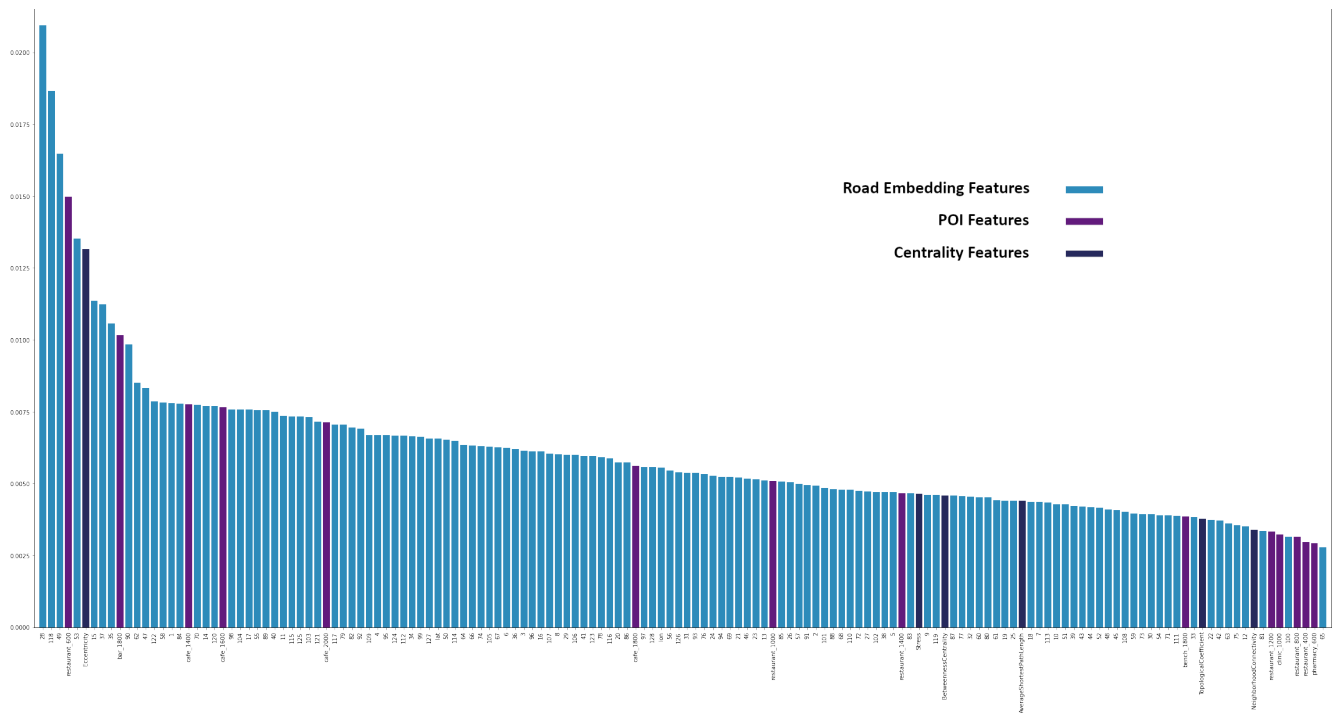


Figure 6. Top 150 features. While a noticeable difference is observed among the top 13 features, we can observe a steady trend among most embedding features. Embedding feature importance account for most of the feature importance. We can also observe that despite having the highest number of features (530) only a small number of POI features appear in the top 150 features. (Bar plot generated through matplotlib²² Version 3.4.3 from matplotlib.org.)

Conclusion

We believe ALF-Score and its various extensions such as ALF-Score+ and ALF-Score++ can be very beneficial and act as very powerful tools for many people from various backgrounds working on different domains. Although ALF-Score can produce results specific to various parameters, such as demographics to provide personalized walkability scores, ALF-Score’s pipeline takes a generalized approach instead to allow for various issues that may not be related to walking or walkability be addressed using this method. For instance, bikeability, school friendliness, transit friendliness, or even POI specialties based on different demographics and perceptions. Moreover, the pipeline may be capable of handling wide variety of features as well as other types of networks instead of road network. For example, subway networks. At its core, ALF-Score requires a vector of user ground truth labels alongside a list of features. ALF-Score uses our dedicated web-tool to collect the ground truth labels and processes them through GLEPO to reflect relative to absolute conversion within a small group of users. However, ALF-Score’s pipeline follows a black box system and works with any compatible input data regardless of how they were prepared. The ground truth data can be processed according to individual researchers’ needs and this step can be bypassed in the pipeline. Although walkability scores generated by ALF-Score and its extensions rely on road network data, the generalization offered by their pipeline can be further distilled to beyond road networks. Road network data is treated as any other features and can be replaced with an appropriate feature based on the issue at hand and the research requirements. We genuinely believe that ALF-Score opens the door to many possibilities well beyond the scope covered in this research.

Methods

ALF-Score++ is the second extension of ALF-Score pipeline with a focus on transferability. ALF-Score++ pipeline 1 utilizes a map database that contains road network data as well as POIs extracted from Statistics Canada²⁴ and OpenStreetMap (OSM)²⁵ respectively. The map database feeds into two separate processes: 1) GIS feature extraction, 2) user data extraction through a web-tool interface. The GIS feature extraction process extracts and generates the required features such as node lists, edge lists, various centrality measures, road embedding, and various POI features. The output of this process is fed into the machine learning component as one of its 3 main input feature sets. User data extraction process involves a web-tool interface that utilizes road data to feature various points on an interactive map where users provide their opinion and data. User data is broken into two separate processes, each of which will result in a separate input to the machine learning component. The

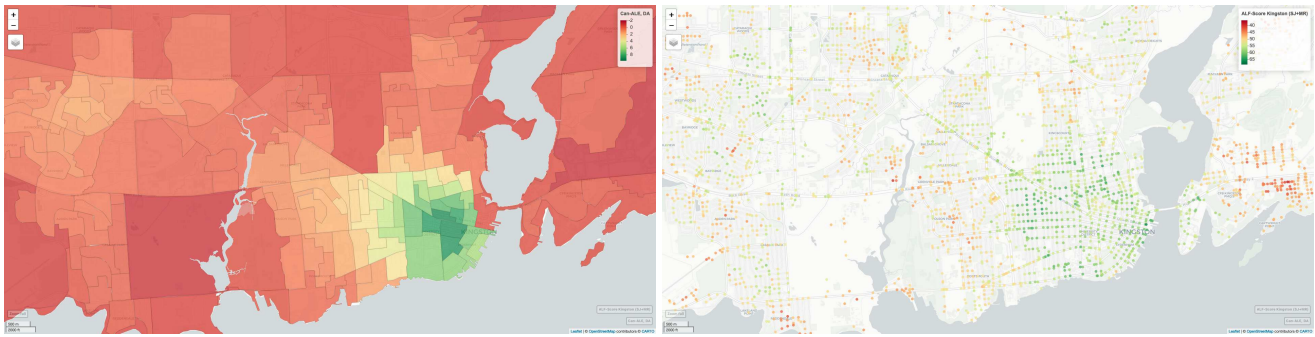


Figure 7. Left: Can-ALE for the city of Kingston, ON. Right: walkability results produced by ALF-Score++ for the city of Kingston, ON using a zero-user-input approach of a model trained through transfer learning based on user data from two cities of St. John’s and Montréal. The road network for Kingston maintains over 3400 nodes. ALF-Score’s walkability scores range between 0-100 units. This range can be adjusted if needed. (Maps generated through RStudio²³ Version 1.2 using mapview package from rstudio.com.)

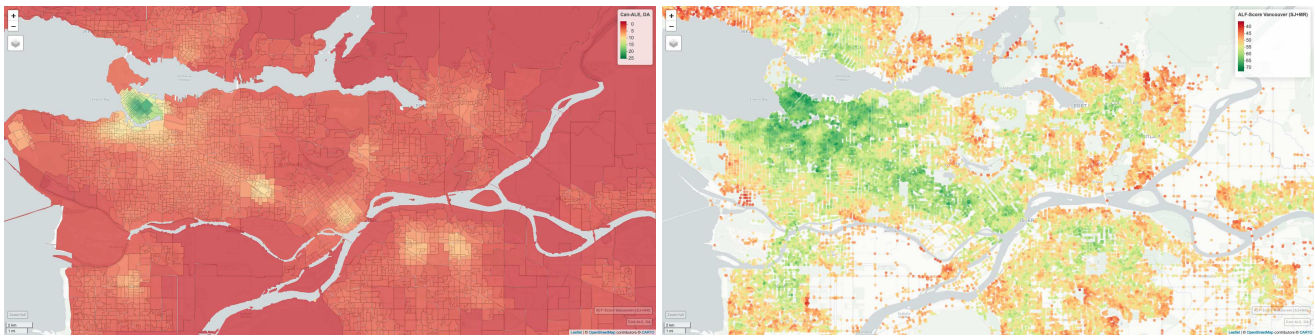


Figure 8. Left: Can-ALE for the city of Vancouver, BC. Right: walkability results produced by ALF-Score++ for the city of Vancouver, BC using a zero-user-input approach of a model trained through transfer learning based on user data from two cities of St. John’s and Montréal. The road network for Vancouver maintains over 45 thousand nodes. (Maps generated through RStudio²³ Version 1.2 using mapview package from rstudio.com.)

276 first process is the collection of user opinions through the web-tool where users provide relative ranks for various points on the map. This process passes the user opinion to our Generalized Linear Extension of Partial Orders or GLEPO algorithm to convert users’ relative ranks to a globalized rank among all submissions. The output of GLEPO is fed to the machine learning component as it’s second feature set. This input serves as the y label during the training and testing processes. The second process of user data revolves around their demographics. This process uses various clustering techniques and unsupervised learning methods to generate a profile clusters. These profile clusters represent users deemed by the algorithm as similar. These profile clusters are then fed into the machine learning component as its third feature set. The machine learning component utilizes these three feature sets in conjunction with it’s internal transfer learning process and the general flow is as follows. GIS features form a feature set which are then associated with specific locations that have their ranking available through the GLEPO algorithm as its y label in the form of {features, label}: {x, y}, where x represent an entry of features. The expectation from the trained models is that they will produce a prediction given an unlabelled set such as {features, ?}: {x, ?} where ? would be replaced with y’ prediction. These models will be trained on the data from only one specific city. The first round of models trained through a deep neural network technique will then be used to transfer their knowledge to the second round of training where transfer learning utilizes appropriate layers while replacing the output layer. The new data used in the transfer learning process will then include features and user opinion from a second city. The output will be a more generalized model capable of transferring its knowledge to cities never seen during its training process. The personalization process utilizes this transfer learning approach to do the same task but on each separate profile cluster, resulting in multiple models capable of predicting personalized walkability scores for cities seen or never seen by the algorithm.

294 Data Preparation

295 To prepare the map database, the first step is to gather the feature set that includes various information such as POI, road embedding and road network data. The POI data is available freely through OpenStreetMap (OSM)²⁵. We utilized Overpass-

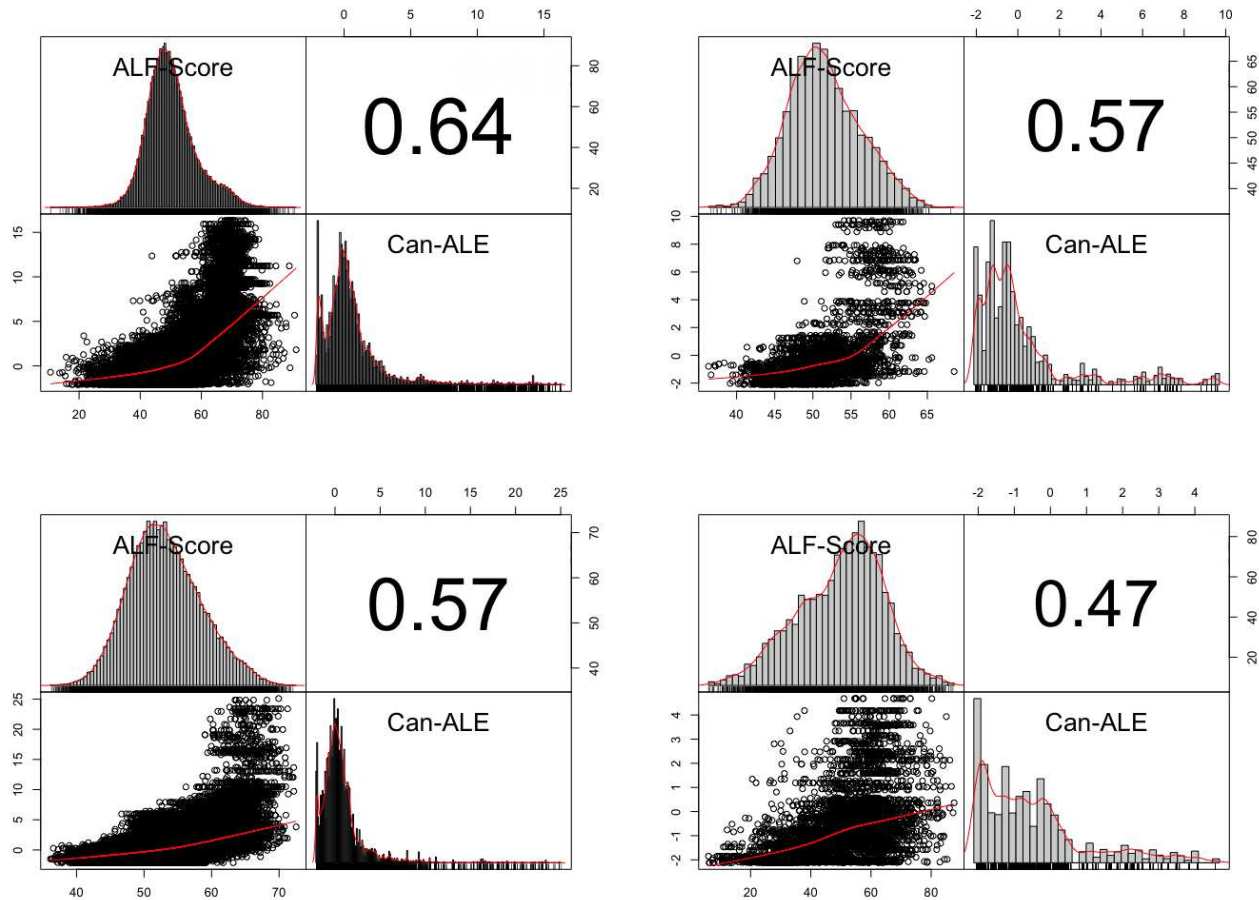


Figure 9. Correlation between ALF-Score++ and Can-ALE for four different cities. Top left: Montréal QC, Top right: Kingston ON, Bottom left: Vancouver BC, Bottom right: St. John's NL. (Correlation figures generated through RStudio²³ Version 1.2 using PerformanceAnalytics package from rstudio.com)

297 Turbo²⁶, with the help of a customized extraction code, to extract OSM POIs from 53 unique amenity categories. Once
 298 complete, we devised a new algorithm that creates POI-based features for all nodes within the network. Below is an example of
 299 a single POI contained within a GeoJSON file extracted from OSM through Overpass-Turbo. Each POI point is divided into
 300 2 parts: 1) description, 2) geometry. Description contains the type and properties of the point while the geometry contains
 301 location's type as well as its coordinates:

```
302 • "type": "Feature", "properties": {"@id": "node/1401297904",
303   "amenity": "fire_station", "name": "Caserne 29 Rosemont"
304   "geometry": {"type": "Point", "coordinates": [ -73.5762681, 45.5453509 ]
```

305 As each POI is represented by a node on the road, we assign a value to 10 separate distance ranges which represent the
 306 number of POIs of a specific category within a specific distance range to a specific node. Based on 53 amenity categories, we
 307 can produce a POI feature list containing 530 feature columns and n rows for the number of unique nodes in the road network.
 308 Below is an example of one possible POI feature header structure followed by an example of a single entry for a unique node:

```
309 • node_id lon lat bar_200 bar_400 bar_600 ... bbq_200 bbq_400 bbq_600 ...
310   317 -73.57113438 45.51020696 0 6 12 ... 14 11 12 ...
```

311 While road network data is available freely from both OSM as well as Statistics Canada²⁴, we chose to extract them from
 312 Statistics Canada in the form of ArcGIS Shapefile²⁷. The Shapefile for the entire Canada was extracted for the year 2016 which
 313 is the most recent Census year available at the time of this research. Furthermore, QGIS²⁸ which is “a free and open-source

314 cross-platform desktop geographic information system application that supports viewing, editing, and analysis of geospatial
315 data”, was used to extract road networks specific to four different cities of St. John’s NL, Montréal QC, Vancouver BC and
316 Kingston ON from the single large Shapefile containing the road network for the entire Canada. All of the individual city
317 sub-networks were further processed to build specific node list and edge lists which are used in various locations within the
318 pipeline. We utilize “shp2graph” package²⁹ through R³⁰ to generate node lists and edge lists for our road networks which
319 have been stored in the form of graphs. It is important to mention that depending on the data source and format the coordinate
320 systems may differ. For instance, some formats may present coordinates in UTM³¹ or WGS³² while others may present them in
321 different coordinate systems. Appropriate conversions, where applicable, may be required. As is the case with many researches,
322 it is absolutely crucial to maintain a unified unit of measurement throughout the research to avoid any unwanted disasters³³.
323 Individual city sub-networks are also processed to generate various complex networks features such as different centralities
324 as well as road embedding features for all road networks. Furthermore, when working with large networks, graph reduction
325 and reconstruction techniques³⁴ may be applicable. Additionally, the edge list for each city is processed through Cytoscape³⁵
326 which is “an open source bioinformatics software platform for visualizing molecular interaction networks and integrating with
327 gene expression profiles and other state data”, to generate a list of network features. To generate road embedding features,
328 edge lists are processed through Node2vec³⁶ which is “an algorithm to generate vector representations of nodes on a graph”.
329 All features that are not numerical go through an encoding processed called one-hot encoding to prepare the features for our
330 machine learning processes.

331 The next step needed to take to prepare the pipeline is the application of Generalized Linear Extension of Partial Orders
332 or GLEPO¹³. GLEPO requires a few data sets such as user opinion, node list and a distance matrix connecting all the nodes
333 within the network. The overall GLEPO pipeline involves multiple algorithms such as *seperateBySub* which is used to prepare
334 user opinions into subsets that are suitable for processing. Various other algorithms such as *calculateDistance*, *FindDistance*,
335 *addToSorted*, *FindVLink*, *RandomizeInsertion*, *normalize* and *GLEPO* are also used to further process user opinion and to
336 convert their relative rankings into generalized scores which are globalized among all opinions. The output of the GLEPO
337 pipeline is a generalized list of user opinion which can be fed into the next pipeline. This globalized list is crucial to the entire
338 structure of ALF-Score as it plays the important role of ground truth used in the machine learning component.

339 Experiments

340 There are three main experimentation scenarios used to guide this research forward, and they are: 1) Matching approach, 2)
341 Combined approach, 3) Zero-user-input approach.

342 **Matching approach** is a scenario where user opinion from a specific city is used to train and test models for the matching
343 city. This approach is an important base to our machine learning pipeline and focuses on testing the feasibility and accuracy of
344 the pipeline derived from user opinion and feature set belonging to the same city. For instance, using user opinion and feature
345 set collected for the city of St. John’s, NL to train and test models on St. John’s. Furthermore, this approach is used for testing
346 the scalability of the model to ensure process stability when it comes to very large road features and user opinion data.

347 **Combined approach** is an approach that focuses on transferability of ALF-Score++’s pipeline. This approach uses data
348 from multiple cities to train and test models. These models can then be applied to cities either seen by the pipeline through
349 the training process or cities never seen by the algorithms before. This approach aims to test and verify that transfer learning
350 can improve the overall generalization of the models while broadening models’ applicability. There are multiple variations
351 in this scenario, specifically how training and test sets are selected. Two of our commonly used variations are random and
352 semi-random selections. In the random selection, a typical 80-20% training-test distribution is used that includes data from two
353 cities. In the semi-random approach, 50% of the data for only one city is randomly selected for testing purposes whereas the
354 remaining 50% is combined with the entire data from the second city to form the training set. The model is tested on both cities.

355 **Zero-user-input approach** aims to use models that are previously trained on specific city/cities to predict walkability
356 scores of other cities. This approach takes advantage of predefined features and pre-trained models to generate walkability
357 scores for points in cities never seen by the algorithms. This approach is very important to help us identify how applicable and
358 transferable are the pre-trained models to data from unseen cities and whether the patterns observed and learned in different
359 cities are similar and transferable to one another. Models in this scenario could have been trained on either a single city or be
360 multi-city models. The models in this scenario can be applied to data from either cities never used in the training process or
361 previously trained cities, making them very versatile.

362 Transfer Learning

363 ALF-Score¹³ pipeline has tested for various supervised and semi-supervised approaches and methods. However, the most
364 promising shallow models are random forest, support-vector machine (SVM) and decision tree whereas the most promising
365 deep model was multi-layer perceptron neural network (MLP). All of these methods generated reasonable accuracy results
366 while random forest performing the best among all. We set up random forest with 100 estimators (the number of trees in the

forest) while its maximum depth of the tree was not limited. Most other parameters such as the number of jobs to run in parallel, the number of features to consider when looking for the best split and bootstrap sampling were set to scikit-learn³⁷ default parameters. Random forest is an ensemble approach. Ensemble learners aim to use multiple weak learners to build a strong learner that perform very well taking a divide and conquer approach. Random forest uses standard decision tree which could be considered as its weak learner. Multiple of these trees will then form a forest which can perform better as a group. Table 2 shows the difference in error between random forest using 100 weak learners as well as a single decision tree. Random forest performs significantly better. There are two specific functions in scikit-learn’s random forest that although not specifically labeled as transfer learning approaches, are geared toward transferring previously learned knowledge. These functions are *warm_start* and *partial_fit*. Warm start aims to fit an estimator repeatedly over the same data set but with varying parameters. Using this approach one can look at various parameters to improve performance while reusing the model learned from previous parameters to save computing resources and time. Warm start is typically used for fine tuning the model parameters. Partial fit on the other hand aims to provide an online machine learning approach while maintaining a fixed model parameters between calls, by allowing for new data in every call. This data is called mini-batch. Online machine learning is a method used to update the predictor in a sequential order as new data becomes available. This is the opposite approach taken in batch learning where the training data set never changes.

Furthermore, MLP was used as a way to utilize deep learning specifically as a doorway to transfer learning. In this paper, we work with transfer learning under the assumption that previously trained models of similar task are available (through ALF-Score). The first step to initiate the transfer learning process is to import three sets of data: 1) previously trained MLP models, 2) GIS features such as POI, centrality and embedding features associated with the new city, 3) user data such as user opinion and demographics associated with the new city. After successful import of data, the usual data processing and preparation steps will need to be taken such as dealing with incomplete entries and processing features through one-hot encoding, where applicable. In this research we use TensorFlow³⁸ to facilitate MLP training and transfer learning processes. TensorFlow is “a free and open-source software library for machine learning and artificial intelligence” that enables us to apply various techniques with very efficient implementations. To set up TensorFlow for transfer learning, the first step is to create a Sequential model. Next we can add multiple Dense layers as our hidden layers. Each dense layer takes in a unit value and an activation function. The unit value which is a positive integer defines the dimensionality of the output space. The activation function³⁹ acts as a trigger based on the input values and fires only if input exceeds a set threshold. In this setup, we use ReLU activation function⁴⁰. If the input is negative, ReLU returns 0, otherwise it will return the actual input. For the last layer that acts as our output layer, the unit is set to 1. It is common to see Softmax activation function being used in classification tasks for the last dense layer, however, since our task is a regression problem we use linear activation function. At this point, the model needs to be compiled with the loss function, optimizer and metrics set. We set the loss function to *mean_absolute_error*, the optimizer to *adam* and the metrics to *mean_squared_error*. The last step is to fit the model by passing the feature set followed by the labels and setting the number of epochs and the size of the validation split. Depending on the batch size, number of epochs and the size of data, the process may take a while. This process will result in a model trained on the {features, label}: {x, y} set. In our approach, we only import the models previously trained through this approach.

Table 6. Various deep neural network settings under which MLP and transfer learning were experimented with.

<i># of Dense Layers</i>	<i>Output Shape Range</i>	<i>Total Parameters</i>	<i>Optimizer</i>	<i># of Epochs</i>
2	8-16	10,945	Adam	200
5	50-300	418,301	Adam	300
11	50-1,000	2,673,301	AdaMax	400
12	50-800	2,303,001	AdaMax	600

ALF-Score uses various combination of dense layers and number of neurons. Table 6 shows a brief set of example settings we have experimented with. To transfer the model generated/imported as above, the first step is to create a new Sequential model and copy the hidden layers desired from the original model over to the new model. In the process we will exclude the output layer. We also need to ensure all transferred layers are frozen by setting them as non-trainable so the algorithm will not modify them. Next, we add a dense output layer to the new model with unit set to 1 and activation function set to linear. Finally we set the loss function to *mean_absolute_error*, the optimizer to *adam* and the metrics to *mean_squared_error* and compile and fit the new model. After a few iterations/epochs, we can try to unfreeze the reused hidden layers to allow back propagation to modify and fine-tune them and re-evaluated the performance. It is also suggested²¹ to reduce the learning rate to avoid changes in weights that are fine-tuned when these layers are unfrozen. A good rule of thumb is to train the model for the new task for a few epochs while the reused layers are frozen. Then unfreeze the reused layers and continue to train, with reduced learning rate, for further fine-tuning these layers. When talking about transfer learning, learning rate is always an

413 important variable to consider. If the learning rate is set too high, training may diverge and if the learning rate is set too low, the
414 processing speed will be very slow to reach a convergence. Experimenting with various parameters may be a good approach to
415 find the best setting that may be most appropriate in a particular task.

416 It is important to note that if the input data of the new task does not have the same shape structure as the data used in the
417 original task, they will need to be processed to match the original size. However, this is not the case with ALF-Score++ since
418 the structure of feature sets used for training various models remain the same. Additionally, according to Géron²⁰ “...transfer
419 learning will work best when the inputs have similar low-level features”. To further expand on this, high-level features are
420 typically more useful in a new task and are generally found in the upper hidden layers. They may be useful to their original
421 task but likely there may not be much relevancy between upper layers from the original task with the new task. So it is a good
422 idea to replace these layers for the new task as they will likely be very different to that of the original task. This however, is
423 not always the case and varies from task to task. For example, a voice recognition task will still need to produce the correct
424 and valid words associated to its output layer. However, top layers may need to recognize words spoken by different people.
425 In this case, reusing the top layers may be more useful²¹. Furthermore, typically the output layer of the original model will
426 be replaced since it is no longer useful as we seek to update the output using the new input. It should be noted here that it is
427 suggested that the more similar the tasks are, the more hidden layers may be used. For instance, in case of ALF-Score++ since
428 the original task is very similar to the new one, we can try by keeping all hidden layers and only replace the output layer.

429 References

- 430 1. Hajna, S., Ross, N. A., Griffin, S. J. & Dasgupta, K. Lexical neutrality in environmental health research: Reflections on the
431 term walkability. *BMC Public Health* **17**, 1–4 (2017).
- 432 2. Forsyth, A. What is a walkable place? The walkability debate in urban design. *Urban Design International* **20**, 274–292
433 (2015).
- 434 3. Gauvin, L. *et al.* From walkability to active living potential: an “ecometric” validation study. *American Journal of*
435 *Preventive Medicine* **28**, 126–133 (2005).
- 436 4. Grant, B. Getting to great places: how better urban design can strengthen San Jose’s future. *Spur Report. San Jose* (2013).
- 437 5. Public Health Ontario. Walkability. [https://www.publichealthontario.ca/en/BrowseByTopic/HealthPromotion/Pages/
438 Walkability.aspx](https://www.publichealthontario.ca/en/BrowseByTopic/HealthPromotion/Pages/Walkability.aspx) (2018).
- 439 6. Ewing, R. & Cervero, R. Travel and the built environment: a meta-analysis. *Journal of the American planning association*
440 **76**, 265–294 (2010).
- 441 7. Spoon, S. *What defines walkability: Walking behavior correlates* (2005).
- 442 8. Colley, R. C., Christidis, T., Michaud, I., Tjepkema, M. & Ross, N. A. An examination of the associations between
443 walkable neighbourhoods and obesity and self-rated health in Canadians. *Heal. Reports* **30**, 14–24 (2019).
- 444 9. Herrmann, T. *et al.* A pan-Canadian measure of active living environments using open data. *Heal. Reports* **30**, 16–26
445 (2019).
- 446 10. Colley, R. C., Christidis, T., Michaud, I., Tjepkema, M. & Ross, N. A. The association between walkable neighbourhoods
447 and physical activity across the lifespan. *Heal. Reports* **30**, 3–14 (2019).
- 448 11. Fry, R., Akbari, A., Mah, S. & Ross, N. Measuring Active Living Environments: An international comparison between
449 Canada and Wales. *Int. J. Popul. Data Sci.* **3** (2018).
- 450 12. Canadian Urban Environmental Health Research consortium (CANUE). *Canadian Active Living Environments Database*
451 *(Can-ALE)* (2016 (accessed May 3, 2020)). <https://canue.ca/data/>.
- 452 13. Ali M. S. Alfosool, Yuanzhu Chen & Daniel Fuller. ALF-Score - a Novel Approach to Build a Predictive Network-Based
453 Walkability Scoring System, DOI: [10.31219/osf.io/tcgqp](https://doi.org/10.31219/osf.io/tcgqp) (2021). DOI: 10.31219 osf.io/tcgqp.
- 454 14. Ali M. S. Alfosool, Daniel Fuller & Yuanzhu Chen. ALF-Score+ - Personalization of a Predictive Network-Based
455 Walkability Scoring System, DOI: [10.31219/osf.io/qdtfj](https://doi.org/10.31219/osf.io/qdtfj) (2021). DOI: 10.31219 osf.io/qdtfj.
- 456 15. Torrey, L. & Shavlik, J. Transfer learning. In *Handbook of research on machine learning applications and trends:
457 algorithms, methods, and techniques*, 242–264 (IGI Global, 2010).
- 458 16. Tan, C. *et al.* A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, 270–279
459 (Springer, 2018).
- 460 17. Zhuang, F. *et al.* A comprehensive survey on transfer learning. *Proc. IEEE* **109**, 43–76 (2020).

- 461 18. Lu, J. *et al.* Transfer learning using computational intelligence: A survey. *Knowledge-Based Syst.* **80**, 14–23 (2015).
- 462 19. Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**, 1–40 (2016).
- 463 20. Géron, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build*
464 *intelligent systems* (O'Reilly Media, 2019).
- 465 21. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).
- 466 22. Hunter, J. D. Matplotlib: A 2d graphics environment. *Comput. Sci. & Eng.* **9**, 90–95, DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) (2007).
- 467 23. RStudio. *RStudio | Open source & professional software for data science teams* ((accessed May 3, 2020)). <https://www.rstudio.com/>.
- 468
- 469 24. Statistics Canada. Statistics Canada: Canada's national statistical agency (2020 (accessed May 3, 2020)). <https://www.statcan.gc.ca/eng/start>.
- 470
- 471 25. OpenStreetMap contributors. *OpenStreetMap* (2020 (accessed May 3, 2020)). <https://www.openstreetmap.org>.
- 472 26. Raifer, M. *Overpass turbo*. Accessed Oct 2021, <http://overpass-turbo.eu/>.
- 473 27. Environmental Systems Research Institute, I. *What is a shapefile?* Accessed Oct 2021, <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm>.
- 474
- 475 28. QGIS Development Team. *Quantum Geographic Information System (QGIS)*. Open Source Geospatial Foundation (2020
476 (accessed May 3, 2020)).
- 477 29. Lu, B. *shp2graph: Convert a SpatialLinesDataFrame Object to an 'igraph'-Class Object* (2018). R package version 0-5,
478 <https://CRAN.R-project.org/package=shp2graph>.
- 479 30. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna,
480 Austria (2018). <https://www.R-project.org/>.
- 481 31. Survey, U. G. *What does the term UTM mean? Is UTM better or more accurate than latitude/longitude?* Accessed Oct
482 2021, <https://www.usgs.gov/faqs/what-does-term-utm-mean-utm-better-or-more-accurate-latitude-longitude>.
- 483 32. Geography, G. *World Geodetic System (WGS84)*. Accessed Oct 2021, <https://gisgeography.com/wgs84-world-geodetic-system/>.
- 484
- 485 33. Lilley, S. Lost in translation. In *System Failure Case Studies*, vol. 3 (National Aeronautics and Space Administration,
486 2009).
- 487 34. Ali M. S. Alfosool, Yuanzhu Chen & Daniel Fuller. Road Importance Using Complex-Networks, Graph Reduction &
488 Interpolation. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, 855–859 (2020).
- 489 35. Consortium, C. *Cytoscape: An Open Source Platform for Complex Network Analysis and Visualization*. Accessed Oct
490 2021, <https://cytoscape.org/>.
- 491 36. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD*
492 *international conference on Knowledge discovery and data mining*, 855–864 (2016).
- 493 37. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
- 494 38. Abadi, M. *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems (2015). Software available from
495 tensorflow.org.
- 496 39. Deep AI, I. *What is an Activation Function?* Accessed Oct 2021, <https://deeptai.org/machine-learning-glossary-and-terms/activation-function>.
- 497
- 498 40. Deep AI, I. *What is ReLu?* Accessed Oct 2021, <https://deeptai.org/machine-learning-glossary-and-terms/relu>.

499 Acknowledgements

500 We would like to acknowledge Wireless Networking and Mobile Computing Laboratory (WineMocol) and Built Environment
501 and Active Populations (BEAP) Lab team members for their contribution with data collection.

502 Author contributions statement

503 Y.C. and D.F. devised the research project. A.M.S.A., Y.C. and D.F. performed the research. A.M.S.A., Y.C. and D.F. analyzed
504 the data and the results. All authors reviewed the manuscript.

505 **Additional information**

506 Correspondence and requests for materials should be addressed to Y.C.