Supplementary Material for A Priority Map for Vision-and-Language Navigation with Trajectory Plans and Feature-Location Cues

Jason Armitage
University of Zurich
Switzerland
jason.armitage@uzh.ch

Leonardo Impett
University of Cambridge
UK
li222@cam.ac.uk

Rico Sennrich
University of Zurich
Switzerland
sennrich@cl.uzh.ch

1. Notation

Notations used in multiple sections of this paper are defined here for fast reference. Auxiliary tasks (ϕ_1, ϕ_2) and the main VLN task ϕ_{VLN} constitute the set of tasks Φ. Inputs and embeddings are specified as l (linguistic), v (visual), and η (multimodal). A complete textual instruction is denoted as τ, ς is a span, and ψ is a perspective. Linguistic and visual inputs for the PM-VLN are denoted as $(\imath'_t, \psi_t)$ and embeddings processed in prioritisation operations are (e_l, e_v). In contrast, U denotes a set of embeddings from the main model, which are derived from inputs $(\bar{e}_\eta, \psi_{cat})$. The notations Δ and $\bigoplus$ are respectively visual boost filtering and self-attention operations. Table 1 provides a reference source for standard notation appearing throughout this paper. Other notations are defined in the sections where they are used.

Notation	Usage in this paper
A	Matrix
AA	Identity matrix
B, b	Bias
$\mathcal{D}$	Dataset
$Train, Dev, Test$	Dataset partitions
$\exists$	Exists
$\forall$	For every (eg member in a set)
g	Function
H	Hypothesis
$\mathcal{L}$	Layer of a model
len	Length
μ	Mean
n	Number of samples
ν	Or
P	Probability
q	Algorithm
S	Signal detected
σ	Standard deviation
Θ	Set of parameters
W, w	Set of weights
$\|x\|$	Sequence
$\triangleq$	Equal by definition

Table 1: Reference List for Standard Notation.

2. Additions to the Method Section

2.1. Theoretical Basis for Cross-modal Prioritisation

This section provides a theoretical basis for a hierarchical process of cross-modal prioritisation that optimises attention over linguistic and visual inputs. In this section we use q to denote this process for convenience. During the main task ϕ_{VLN}, q aligns elements in temporal sequences τ and $Route$ and localises spans and visual features w.r.t. a subset of all entities Ent in the routes:

$$q = \|x_l - x_v\| \underset{subject\,to}{\rightarrow} maxP_{D_{Ent}}[\tau, Route] \leq R \tag{1}$$

Inputs in ϕ_{VLN} consist of a linguistic sequence τ and a visual sequence $Route$ for each trajectory j in a set of trajectories. As a result of the process followed by Chen *et al*. [4] to create the Touchdown task, these inputs conform to the following definition.

Definition 1 (Sequences refer to corresponding entities). *At each step in j, $|x_l|$ and $|x_v|$ are finite subsequences drawn from τ_j and $Route_j$ that refer to corresponding entities appearing in the trajectory $ent_j \subset Ent$.*

In order to simplify the notation, these subsequences are denoted in this section as x_l and x_v. Touchdown differs from other outdoor navigation tasks [5] in excluding supervision on the alignment over cross-modal sequences. Furthermore $len(\tau_j) \neq len(Route_j)$ and there are varying counts of subsequences and entities in trajectories. In an approach to ϕ_{VLN} formulated as supervised classification, an agent's action at each step $\alpha_t \equiv$ classification $c_t \in \{0, 1\}$ where c is based on corresponding ent_t in the pair $(x_l, x_v)_t$. The likelihood that c_t is the correct action depends in turn on detecting S signal in the form of ent_t from noise in the inputs. The objective of q then is to maximise P_S for each point in the data space.

The process q is composed of multiple operations to perform two functions of high-level alignment g_{Align} and localisation g_{Loc}. At the current stage stg, function g_{Align} selects one set of spans $\varphi_{stg} \in (\varphi_1, \varphi_2, \ldots, \varphi_n)$

where $stg \begin{cases} Start,\ if\, t = 0 \\ End,\ if\, t = -1 \\ \forall\ stg_{other},\ n \in N \in \sum_{n=1}^{n_1} > t_{-1}\ otherwise. \end{cases}$

This is followed by the function g_{Loc}, which predicts one of $\varsigma_{scnt_0} \vee \varsigma_{scnt_{0-1}}$ as the span ς relevant to the current trajectory step $scnt$

where $scnt \begin{cases} scnt_0,\ if\, (\tau, \psi_t) = 0 \\ scnt_{0-1},\ otherwise. \end{cases}$

We start by describing the learning process when the agent in ϕ_{VLN} is a transformer-based architecture $Enc + Clas$ excluding an equivalent to q (*e.g*. VisualBERT in Table 1 of the main report). $Enc{+}Clas$ is composed of two core subprocesses: cross-modal attention to generate representations $q(\bigoplus(L \Longleftrightarrow \widetilde{V}))$ and a subsequent classification $Clas(\widetilde{e_\eta}')$.

Definition 2 (Objective in $Enc + Clas$). *The objective $Obj_1(\theta)$ for algorithm $q(\bigoplus(L \Longleftrightarrow \widetilde{V})$, where L and V are each sequences of samples $\{x_1, x_2, \ldots, x_n\}$, is the correspondence between samples x_l and x_v presented at step t in $\sum_{i=1}^{n} t_i = t_1 + t_2, \ldots + t_n$.*

It is observed that in the learning process for $Enc{+}Clas$, any subprocesses to align and localise finite sequences x_l and x_v w.r.t. ent_j are performed as implicit elements in the process of optimising $Obj_1(\theta)$. In contrast the basis for the hierarchical learning process enabled by our framework FL_{PM} - which incorporates q_{PM} with explicit functions for these steps - is given in Theorem 1.

Theorem 1. *Assuming x_l and x_v conform to Definition 1 and that $\forall\, x \in L\, \exists\, x \in V$, an onto function $g_{Map} = mx{+}b, m \neq 0$ exists such that:*

$$g_{Map}(x_l, x_v) \rightarrow max\left[ent_j^{(x_l,x_v)} \in Ent\right] \tag{2}$$

In this case, additional functions g_{Align} and g_{Loc} - when implemented in order - maximise g_{Map}:

$$max\ P_{S_{ent_j}} = max\ g_{Map}(x_l, x_v) \underset{subject\,to}{\rightarrow} (\overrightarrow{g_{Align}, g_{Loc}, g_{Map}})\ \forall\, ent_j^{(x_l,x_v)} \in L_j \cap V_j \tag{3}$$

Remark 1 *Let $P(max\ g_{Map})$ in Theorem 1 be the probability of optimising g_{Map} such that the number of pairs $N^{(x_l,x_v)}$ corresponding to $ent_j \in L_j \cap V_j$ is maximised. It is noted that $N^{(x_l,x_v)}$ is determined by all possible outcomes in the set of cases $\{(x_l, x_v) \Leftrightarrow ent_j,\ (x_l, x_v) \not\Leftrightarrow ent_j,\ x_l \not\Leftrightarrow x_v\}$. As the sequences of instances i in x_l, x_v and ent_j are forward-only, it is also noted that $N_{t+1}^{(x_l,x_v)} < N_t^{(x_l,x_v)}$ if $ent_i \notin x_{li}$, $ent_i \notin x_{vi}$, or $ent_i^{x_l} \neq ent_i^{x_v}$. By definition, $N_{t+1}^{(x_l,x_v)} > N_t^{(x_l,x_v)}$*

if $P(ent_i = x_{li} = x_{vi})$ - where the latter probability is s.t. processes performed within finite computational time $CT(n)$ - which implies that $P(max\ g_{Map})|P(ent_i = x_{li} = x_{vi})$.

Remark 2. *Following on from Remark 1, $CT(n^{P(ent_i=x_{li}=x_{vi})})$ when q contains g_t, and function $g_t(max(N^{(x_l,x_v)} \Rightarrow ent_j \in L_j \cap V_j)$, where $g_t \in G < CT(n^{P(ent_i=x_{li}=x_{vi})})$ when q does not contain $g_t < CT(n^{P(ent_i=x_{li}=x_{vi})})$ when q contains g_t, and function $g_t(max(N^{(x_l,x_v)} \not\Rightarrow ent_j \in L_j \cap V_j)$.*

Discussion In experiments, we expect from Remark 1 that results on ϕ_{VLN} for architectures such as $Enc+Clas$ - which exclude operations equivalent to those undertaken by the onto function g_{Map} - will be lower than the results for a framework $\mathrm{FL_{PM}}$ over a finite number of epochs. We observe this in Table 1 of the main report when comparing the performance of respective standalone and + $\mathrm{FL_{PM}}$ for VisualBERT and VLN Transformer systems. Poor results for variants (a) and (h) in Tables 2 and 3 of the main report in comparison to $\mathrm{FL_{PM}}$ + VisualBERT(4l) also support the expectation set by Remark 2 that performance will be highly impacted in an architecture where operations in g_{Map} increase the number of misalignments.

Proof of Theorem 1 *We use below $a*$ for a generic transformer-based system that predicts α on (L, V), ∇x for gradients, and Θ^{a*} to denote $\Theta^{Enc+Clas}\ \nu\ \Theta^{Enc+q}$. Let sequence $x_l = [ent_1, ent_2, \ldots, ent_{n_1}]$ and sequence $x_v = [ent_1, ent_2, \ldots, ent_{n_2}]$, where n_1 and n_2 are unknown. Furthermore at any point during learning, $P_S(x_l, x_v)$ is spread unevenly over ent_j in relation to $\Theta^{a*} \approx \mathcal{X}$.*

Propositions *We start with the case that $\exists\ ent_j : ent^{(x_l)}$ and $ent^{(x_v)}$. $CT(n^{Ent \in L \cap V})$ for $\Theta^{a*+g_t} < CT(n^{Ent \in L \cap V})$ for Θ^{a*} where g_t accounts for $\Delta(Len_1, Len_2)$. We next consider the case where $\nexists\ ent_j : ent^{(x_l)}\ \nu\ ent^{(x_v)}$. $Where \nexists\ g_{Loc}$ $then$ $P_S^{(x_l,x_v)} < \exists\ g_{Loc}\ P_S^{(x_l,x_v)}$. We conclude with the case where $\exists\ Ent : x_l\ \nu\ x_v$. In $P_S^{A*}\ ent^{(x_l)} \bigoplus ent^{(x_v)}$ $when$ $ent^{(x_l)} \neq ent^{(x_v)}$.*

As $(Ent_L, Ent_V) \Rightarrow Ent$, $\Theta^{a*} \approx max(N^{(x_l,x_v)}) \in \mathcal{X}$. $P_S^{(x_l,x_v)}$ where $ent_i = x_{li} = x_{vi} > ent_i \in \Theta^{a*} \approx max(N^{(x_l,x_v)})$. Furthermore $P \exists\ ent \in \approx (ent_i) > \nexists\ ent \not\approx ent_i$. Therefore slope ∇x increases and $CT(n^{Ent \in L \cap V})$ for $\Theta^{a*+q} < CT(n^{Ent \in L \cap V})$.

2.2. Visual Boost Filtering

We provide further description on the initial operations conducted during feature-level localisation. Parameterised visual boost filtering as proposed by Carranza *et al*. [3] is applied to perspectives. Let $Conv_{VBF}$ be a convolutional layer with a kernel κ and weights W that receives as input ψ_t. In the first operation g_{USM}, a Laplacian of Gaussian kernel κ_{LoG} is applied to ψ_t. The second operation g_{VBF} consists of subtracting the output e_v from the original tensor ψ_t:

$$g_{VBF}(e_v) = (\lambda - 1)(e_v) - g_{(USM)}(\psi_t) \tag{4}$$

where λ is a learned parameter for the degree of sharpening.

A combination of g_{USM} and g_{VBF} is equivalent to adaptive sharpening of details in an image with a Laplacian residual [2]. Here operations are applied directly to e_v and adjusted at each update of the convolutional layer with a parameterised control $\beta\lambda$. In the simple and efficient implementation from [3], σ in the distribution $LoG(\mu_j, \sigma_j)$ is fixed and the level of boosting is reduced to a single learned term

$$\Delta z(x_1, x_2) = \beta\lambda(\sum_j (AA'_{\kappa_{i_j}} - A_{W_{\kappa_{i_j}}})_z) \tag{5}$$

where A_W is a matrix of parameters and AA' is the identity.

3. Datasets

3.1. Generation and Partition Sizes

The MC-10 dataset consists of visual, textual and geospatial data for landmarks in 10 US cities. We generate the dataset with a modified version of the process outlined by [1]. Two base entity IDs - Q2221906 (“geographic location”) and Q83620 (“thoroughfare”) - form the basis of queries to extract entities at a distance of $<= 2$ hops in the Wikidata knowledge graph[1]. Constituent cities consist of incorporated places exceeding 1 million people ranked by population density based on data for April 1, 2020 from the US Census Bureau[2]. Images and coordinates are sourced from Wikimedia and text summaries are extracted with the MediaWiki API. Geographical cells are generated using the S2 Geometry Library[3] with a range of n entities $[1, 5]$. Statistics for MC-10 are presented by partition in Table 2. As noted above, only a portion of textual inputs are used in pretraining and experiments.

[1] https://query.wikidata.org/

[2] https://www.census.gov/programs-surveys/decennial-census/data/datasets.html

[3] https://code.google.com/archive/p/s2-geometry-library/

	Train	Development
Number of entities	8,100	955
Mean length per text summary	727	745

Table 2: Statistics for the MC-10 dataset by partition.

TR-NY-PIT-central is a set of image files graphing path traces for trajectory plan estimation in two urban areas. Trajectories in central Manhattan are generated from routes in the Touchdown instructions [4]. Links E connecting O in the Pittsburgh partition of StreetLearn [7] are the basis for routes where at least one node is positioned in the bounding box delimited by the WGS84 coordinates (40° 27' 38.82", -80° 1' 47.85") and (40° 26' 7.31", -79° 59' 12.86"). Labels are defined by step count cnt in the route. Total trajectories sum to 9,325 in central Manhattan and 17,750 in Pittsburgh. In the latter location, routes are generated for all nodes with 50 samples randomly selected where $cnt =< 7$ and 200 samples where $cnt > 7$. The decision to generate a lower number of samples for shorter routes was determined by initial tests with the ConvNeXt Tiny model [6]. We opt for a maximum cnt of 66 steps to align with the longest route in the training partition of Touchdown. The resulting training partition of samples for Pittsburgh consists of 17,000 samples and is the resource used to pretrain g_{PMTP} in the PM-VLN module.

3.2. Samples from Datasets

In auxiliary task ϕ_2, the g_{PMF} submodule of PM-VLN is trained on visual, textual, and geodetic position data types. Path traces from the TR-NY-PIT-central are used in ϕ_1 to pretrain the g_{PMTP} submodule on trajectory estimation. Samples for entities in MC-10 and path traces in TR-NY-PIT-central are presented in Figures 1 and 2.

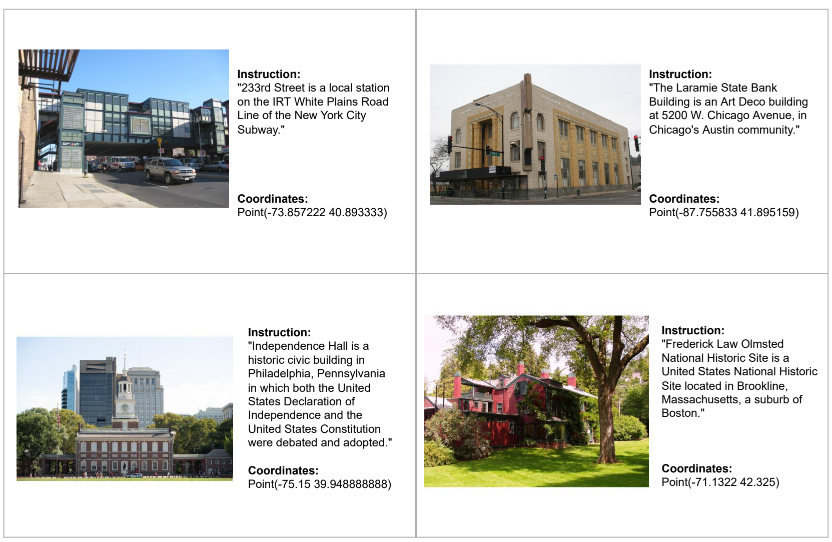

Figure 1: Samples from the MC-10 dataset.

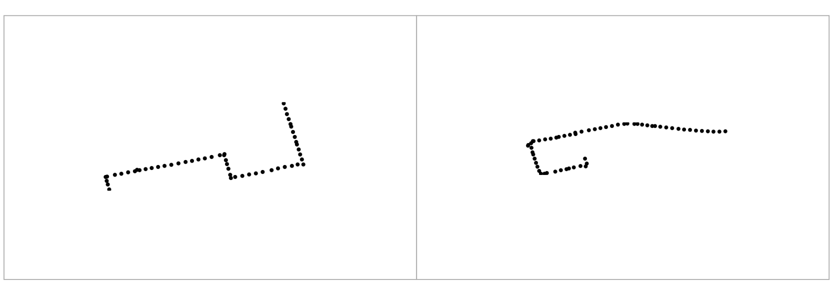

Figure 2: Samples from the TR-NY-PIT-central dataset with path traces representing routes in central Pittsburgh.

4. Code and Data

Source code for the project and instructions to run the framework are released and maintained in a public GitHub repository under MIT license (`https://github.com/JasonArmitage-res/PM-VLN`). Code for the environment, navigation, and training adheres to the codebases released by [8] and [4] with the aim of enabling comparisons with benchmarks introduced in prior work on Touchdown. Full versions of the MC-10 and TR-NY-PIT-central datasets are published on Zenodo under Creative Commons public license (`https://zenodo.org/record/6891965#.YtwoS3ZBxD8`).

5. Acknowledgments

This publication is supported by the Digital Visual Studies program at the University of Zurich and funded by the Max Planck Society. RS has received funding by the Swiss National Science Foundation (project MUTAMUR; no. 176727). The authors would like to thank Howard Chen, Piotr Mirowski, and Wanrong Zhu for assistance and feedback on questions related to the Touchdown task, data, and framework evaluations.

References

[1] Jason Armitage, Endri Kacupaj, Golsa Tahmasebzadeh, Maria Maleshkova, Ralph Ewerth, and Jens Lehmann. Mlm: A benchmark dataset for multitask learning with multiple languages and modalities. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2967–2974, 2020.

[2] B Bayer. A method for the digital enhancement of unsharp, grainy photographic images. *Advances in Computer Vision and Image Processing*, 2:Chapter–2, 1986.

[3] Jose Carranza-Rojas, Saul Calderon-Ramirez, Adán Mora-Fallas, Michael Granados-Menani, and Jordina Torrents-Barrena. Unsharp masking layer: injecting prior knowledge in convolutional networks for image classification. In *International Conference on Artificial Neural Networks*, pages 3–16. Springer, 2019.

[4] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019.

[5] Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. Learning to follow directions in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11773–11781, 2020.

[6] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. *Advances in Neural Information Processing Systems*, 31:2419–2430, 2018.

[8] Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazoo Sone, Sugato Basu, and William Yang Wang. Multimodal text style transfer for outdoor vision-and-language navigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1207–1221, 2021.