What matters in *ImageNav*: architecture, pre-training, sim settings, pose

Gianluca Monaci, Philippe Weinzaepfel, Christian Wolf
NAVER LABS Europe, Grenoble, France

Introduction – State-of-the-art image goal navigation (*ImageNav*) methods either rely on dedicated image-matching [7] or pre-training of vision modules on relative pose estimation [3] or image reconstruction [14]. Recently, findings reported in [11] suggest that *ImageNav* can be solved by very low-capacity ResNet with channel-wise stacking and RL-training alone, without pre-training. These results raise interesting questions: can directional information, crucial to tackle *ImageNav*, be learned by RL alone, and by comparably simple architectures? In this study we investigate the effect of architectural choices like late fusion, channel stacking and cross-attention, and find that:

- **Pre-training and early patch-wise fusion are essential for strong performance**, compared to late fusion.
- Success of recent frugal channel stacking architectures is likely due to a simulator setting allowing agents to slide along obstacles. Interestingly, capabilities learned in this regime can be transferred to realistic settings **if the transfer includes weights of the perception** network.
- **Navigation and (emerging) relative pose estimation performance are correlated**.

Methods – We study the *ImageNav* task in photo-realistic 3D environments, where an agent is given a goal image $\mathbf{g}$ and has to navigate from a starting location to the goal position using only RGB images $\mathbf{o}_t$. All our experiments are done with variants of the same standard agent, which maintains a recurrent episodic memory $\mathbf{h}_t$, integrates the observation $\mathbf{o}_t$ and the goal image $\mathbf{g}$, and predicts actions $\mathbf{a}_t$:

$$\begin{aligned} \tilde{\mathbf{g}}_t &= \phi(\mathbf{o}_t, \mathbf{g}) && \texttt{// binocular encoder} \\ \mathbf{h}_t &= h(\mathbf{h}_{t-1}, \tilde{\mathbf{g}}_t, \zeta(\mathbf{a}_{t-1})) && \texttt{// state update} \\ \mathbf{a}_t &\sim \pi(\mathbf{h}_t), && \texttt{// policy} \end{aligned} \tag{1}$$

where h is the function updating the hidden state $\mathbf{h}_t$ of a GRU [5], ϕ and ζ are trainable encoders, and π is a linear policy trained with PPO [10] with reward as in [3, 4].

We fix h and π, and investigate the impact of various building blocks used in literature for binocular encoder ϕ:

Late fusion Networks such as [1, 14], use separate networks ϕ_o and ϕ_g to encode observation and goal, hence $\phi(\mathbf{o}_t, \mathbf{g}) = [\phi_o(\mathbf{o}_t), \phi_g(\mathbf{g})]$ where [.] denotes concatenation. $\phi_o(\mathbf{o}_t)$ and $\phi_g(\mathbf{g})$ are compared "*late*", which makes it generally harder to be done on a local image level, unless the representations retain sufficient spatial structure.

ChannelCat as in [11] uses a single network to encode both, observation and goal, which are channel stacked into one input image, denoted as $\phi\left([\mathbf{o}_t, \mathbf{g}]_{\text{dim}=1}\right)$, where 0 is the batch dimension and 1 the channel dimension.

SpaceToDepth reshapes image patches into channel values, was introduced in [8] and is used in [11] in combination with ChannelCat. We study whether a ResNet with this module can compute correspondences across large spatial dimensions in one convolutional layer, somewhat reminiscent of cross-attention, only with few parameters.

Cross-attention is a natural way to compute correspondences between local images parts [3], as each patch in one image can be naturally linked to one or more patches in the other through the cross-attention distribution.

Experiments – We train all agents from scratch with the same experimental protocol using the Habitat simulator [9] on 72 train scenes of the Gibson dataset [13], and evaluate by success rate (SR) and SPL [2] on the val split (14 scenes), using the default Habitat episode definition. All images are 112×112 and the discrete action space is $\mathcal{A}$ ={`MOVE FWD 0.25m`, `TURN LEFT` 10°, `TURN RIGHT` 10°, and `STOP`}. An episode is successful if the agent calls `STOP` within 1m of the goal position *and* within its 1000 steps budget.

We test networks ϕ implemented as ResNet9 [1, 11], ViT-Small [12] and DEBiT-Base [3], available in their respective public repositories. ζ embeds the previous action in a 32D feature, h is a GRU with 2 layers of hidden dimension 128, followed by a linear Actor-Critic policy π.

A critical setting in the Habitat simulator is the binary **`Sliding`** switch, which is known to have a big impact on sim2real transfer [6]: when `True`, the agent can slide along obstacles when colliding, against the more realistic behavior of stopping. While this setting is `True` by default, there is consensus in the field that it should be set to `False` to decrease the sim2real gap. All methods we could verify use `Sliding=False`, with the notable exception of [11]. We therefore performed experiments with both settings and observed a big influence of this parameter. Table entries are color-coded into: (i) `Sliding=True` and (ii) `Sliding=False`; Pre-train indicates that ϕ has been pre-trained on relative pose and visibility estimation (RPVE), the default DEBiT setup [3].

Model	s2d[†]	Backbone	SR	SPL	SR	SPL
(a) Late Fusion	✗	ResNet9	13.8	8.0	12.8	7.1
(b) Late Fusion [1]	✓	ResNet9	12.5	7.6	13.2	8.9
(c) Late Fusion	✗	ViT-Small	12.5	6.7	6.9	4.5
(d) ChannelCat	✗	ResNet9	83.2	43.9	44.6	23.4
(e) ChannelCat [11]	✓	ResNet9	83.6	42.1	31.7	18.7
(f) ChannelCat	✗	ViT-Small	71.1	34.3	35.3	16.2
(g) Cross-attn	✗	DEBiT-B	0.0	0.0	0.0	0.0
(h) Cross-attn [3]	✗	DEBiT-B[‡]	**90.5**	**60.3**	**81.7**	**52.0**

Table 1. **Agents with different visual encoders** trained and validated with `Sliding=True` or `Sliding=False`. **s2d**[†]=*SpaceToDepth*. ‡=pre-trained for RPVE.

Results – Table 1 summarizes results for different architectures and settings. With `Sliding=True`, ChannelCat (d)-(f) obtains excellent performance, close to DEBiT-B (h) which has a larger, more complex architecture, and is pre-trained on RPVE. Without pre-training, DEBiT is not exploitable, as also reported in [3]. Late Fusion architectures (a)-(c) underperform, and SpaceToDepth has no significant impact on either ChannelCat or Late Fusion models. With `Sliding=False` the trends change dramatically. While the impact on the (previously already underperforming) Late Fusion architecture is similar, ChannelCat, (d)-(f), now breaks down and performance is halved, or less. In contrast, DEBiT is able to cope well with the more realistic `Sliding=False` setting, arguably because of its strong pre-trained visual encoder, confirming the importance of visual pre-training.

We then investigate whether (i) agents trained in their respective settings (`Sliding=True / False`) have similar capabilities but perform differently due to the difference in task difficulty, or (ii) training with sliding actually leads to different and potentially better performing agents. We test these hypotheses by performing experiments loading the weights of the *ChannelCat+ResNet9* agent (Tab. 1e) trained with `Sliding=True`, and validating it on `False`. Table 2, row (b) shows that this agent achieves SR=54.6%, compared to the baseline of 31.7% trained on `False` (a). This is surprising: some capabilities learned with sliding enabled can be transferred to more realistic settings. Finetuning this agent for 100M steps on `False` provides further gains and yields SR of 65.7% (c).

To pinpoint this effect, we load different parts of the agent trained with `Sliding=True`. Transferring the action part of the agent (ζ, h, π), rows (d) and (e), does not lead to any discernible performance. However, transferring the perception weights ϕ and training the action part, rows (f) and (g), leads to exploitable results. With SR=38.5% for the finetuned version (g), performance are higher than the in-domain baseline of 31.7% trained on `False`.

We conjecture that the easier task (`True`) allows to learn additional capabilities that transfer to the harder task, and which are partially related to perception (since performance in Tab. 2(g) > 2(a)), but also to action, since Tab. 2(b) ≫ 2(a) and 2(b) ≫ 2(g). We hypothesize that training with `False` leads to undertraining of both action and perception: the policy gets stuck (which we empirically confirmed) and does not learn to cope with the last meters of each episode; this, in turn, leads to undertraining the comparison between the (hardly ever seen) goals and observations.

Checkpoint	Perception ϕ	Action ζ	h	π	(%) SR	SPL
(a) Load all “false”	$f*$	$f*$	$f*$	$f*$	31.7	18.7
(b) Load all “true”	$t*$	$t*$	$t*$	$t*$	54.6	27.5
(c) Load all “true”	$t\rightarrow$	$t\rightarrow$	$t\rightarrow$	$t\rightarrow$	**65.7**	**34.1**
(d) Load action “true”	$\circlearrowright$	$t*$	$t*$	$t*$	0.0	0.0
(e) Load action “true”	$\circlearrowright$	$t\rightarrow$	$t\rightarrow$	$t\rightarrow$	6.1	4.8
(f) Load perception ”true”	$t*$	$\circlearrowright$	$\circlearrowright$	$\circlearrowright$	26.4	14.3
(g) Load perception “true”	$t\rightarrow$	$\circlearrowright$	$\circlearrowright$	$\circlearrowright$	38.5	20.3

Table 2. **OOD behavior and cross-domain transfer** f: load from `Sliding=False`, t: load from `Sliding=True`, $*$: frozen, $\rightarrow$: finetune, $\circlearrowright$: re-train from scratch.

Model (Table nr. + row)		s2d[†]	Backbone	S[‡]	%corr.poses 1m,10°	2m,20°	%corr.vis. <0.05
Late Fusion	1b	✓	ResNet9	✓	9.0	29.6	16.1
ChannelCat	1e	✓	ResNet9	✓	18.4	41.6	20.8
Late Fusion	1b	✓	ResNet9	✗	8.7	28.5	16.1
ChannelCat	1e	✓	ResNet9	✗	12.5	31.9	19.2
ChannelCat	2c	✓	ResNet9	→	18.2	41.4	21.1
ChannelCat	2d	✓	ResNet9	→	5.8	22.9	6.7
ChannelCat	2e	✓	ResNet9	→	7.2	26.1	11.9
ChannelCat	2g	✓	ResNet9	→	18.6	41.6	21.0
Cross-attn	*1h*	✗	*DEBiT-B*	*N/A*	*92.1*	*96.8*	*88.8*

Table 3. **Probing RPVE:** *DEBiT-B* is not comparable as it was pre-trained for RPVE. **S**[‡] = `Sliding=True`. The third block shows agents finetuned (→) from `True` to `False`, cf. Tab. 2.

Finally, we investigate how well the different visual encoders can extract directional information, a crucial skill to solve *ImageNav*. The frozen visual encoders ϕ of agents in Tab. 1 are probed with an MLP network of hidden size 1024, trained to predict RPVE using the same procedure, dataset and loss described in [3]. Tab. 3 displays performance measured as % of correct poses for given distance and angle thresholds, and % of predictions within a 0.05 margin of the ground-truth value. While ChannelCat obtains high navigation performance with `Sliding=True`, pose estimation performance remains limited. Late Fusion models achieve low navigation and pose estimation performance. In the `Sliding=False` setting, pose estimation performance drop, especially for ChannelCat. Transferring models trained with `True` on the `False` setting provides gains not only in navigation but also on pose estimation — only when at least the weights of encoder ϕ are transferred, but in particular when the whole agent is transferred and finetuned, further corroborating our conjecture, that the perception model is undertrained when sliding is disabled.

References

[1] Ziad Al-Halah, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[2] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *arXiv preprint*, 2018. 1

[3] Guillaume Bono, Leonid Antsfeld, Boris Chidlovskii, Philippe Weinzaepfel, and Christian Wolf. End-to-End (Instance)-Image Goal Navigation through Correspondence as an Emergent Phenomenon,. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 2

[4] Prithvijit Chattopadhyay, Judy Hoffman, Roozbeh Mottaghi, and Aniruddha Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. In *IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 15691–15700, 2021. 1

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 1

[6] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics Autom. Lett.*, 2020. 1

[7] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to objects specified by images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[8] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[9] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *International Conference on Computer Vision (ICCV)*, 2019. 1

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017. 1

[11] Xinyu Sun, Peihao Chen, Jugang Fan, Jian Chen, Thomas Li, and Mingkui Tan. Fgprompt: fine-grained goal prompting for image-goal navigation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[12] Ross Wightman. Pytorch image models. `https://github.com/huggingface/pytorch-image-models`, 2019. 1

[13] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[14] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav. In *arXiv preprint*, 2023. 1