Lab Log

Noah Syrkis

September 7, 2025

1 | August 27th, 2025

- Gemma did 8k intel pieces in 90 sec. on V100
- Successfully computing intel on $\frac{n}{\ln(n)}$ units
- Test ran with 280k total units (runs in minutes)
- Opting for generating .gif instead of www?
- *Issue*: plan / bt split too coarse for big map?
- *next*: Add color to gif, and run test 2 step plan

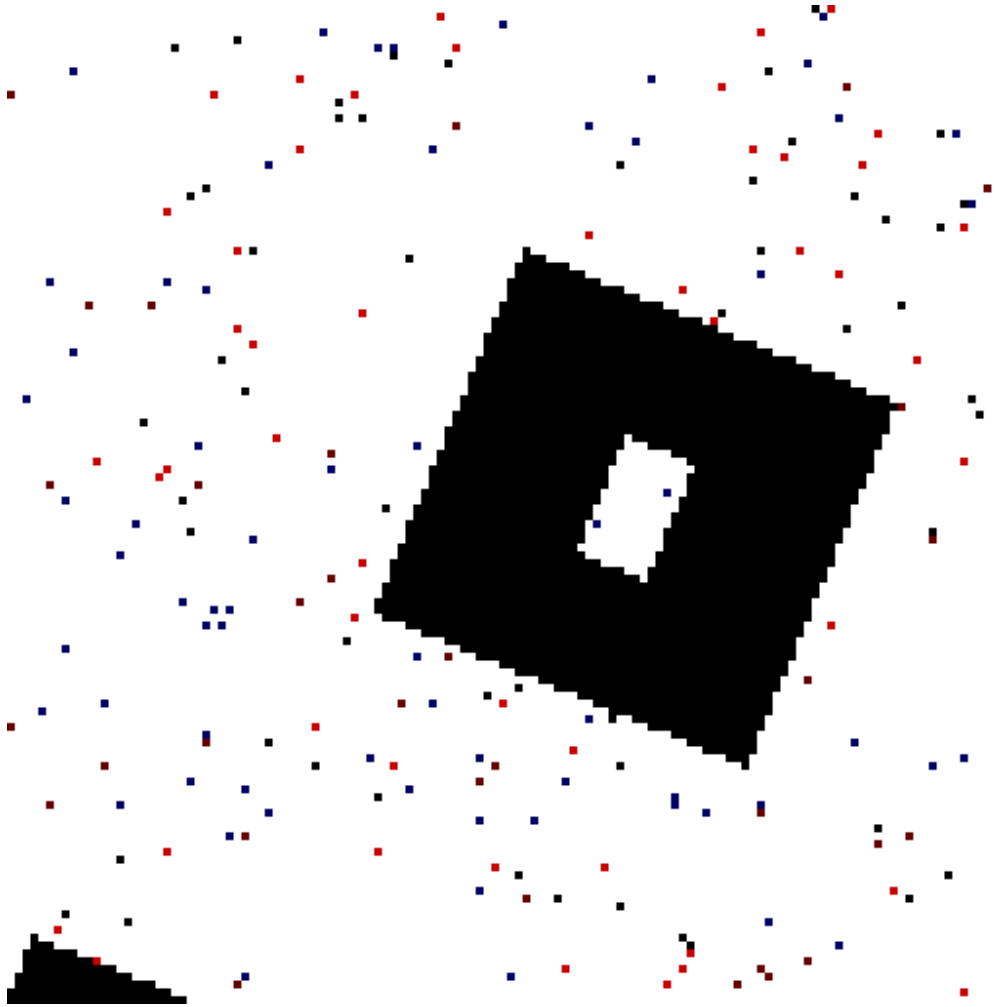


Figure 1: Simulating 8.000 units in 420 by 420 map

2 | Updates

- Gemma3 270M is now working with our code (sent pull request to Google)
- 270M helps a lot during development (and might be enough with fine tuning)
- Now estimating importance *real time* inside simulation loop by
  1. computing imagined trajs based on *all* intel, and
  2. computing many traj baesd on random subset of intel, and
  3. fitting linear model to mask matrix, predicting distance to 1) and *sorting*
- Beginning to reintegrate Nebellum into Svelte frontend (Use APP-6E NATO symbols?)
- Next weeks is just scaling, integrating with front end, debugging, clear examples, etc.
- `pcgym` (modified Togelius `gym-pcgrl`) for lab (and other) use

3 | Gemma

- Comes in 1b, 4b, 12b, and 27b (and all in 4int)
- 99 custom tokens (using `<pos>` and `<hp>`)
- We can now `vmap` across pieces of intel...
- ... and use Gemma inside `jit`, `scan` and `vmap`

```
<start_of_turn>user
{instruction}
{intel}<end_of_turn>
<start_of_turn>model
{response}
```

Listing 1: Gemma intel template

4 | Nebellum

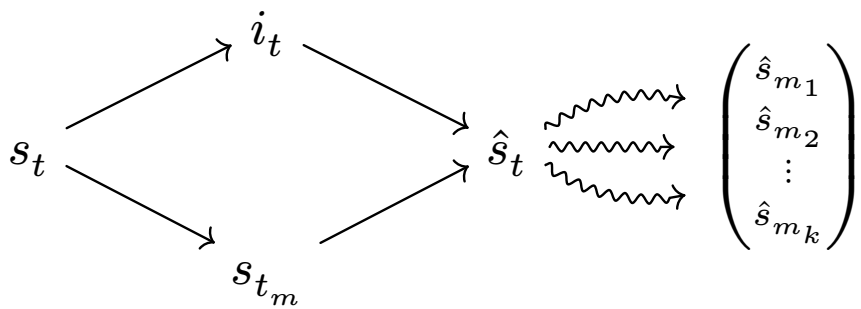


Figure 2: At time $t \in \{1, 2, ..., n\}$ we have state s_t, intel i_t and, masked state s_{t_m}. We combine these into $\hat{s}_t$. Simulating k trajectories $\vec{\hat{s}}_m$, $\hat{s}_t$ is used to estimate the importance of different aspects of $\hat{s}_t$.

- `encode` maps s_t to intel i_t and masked state s_m
- `decode` maps i_t and s_m to estimated state $\hat{s}_t$
- `sample` maps $\hat{s}_t$ to k potential final states $\vec{\hat{s}}_n$

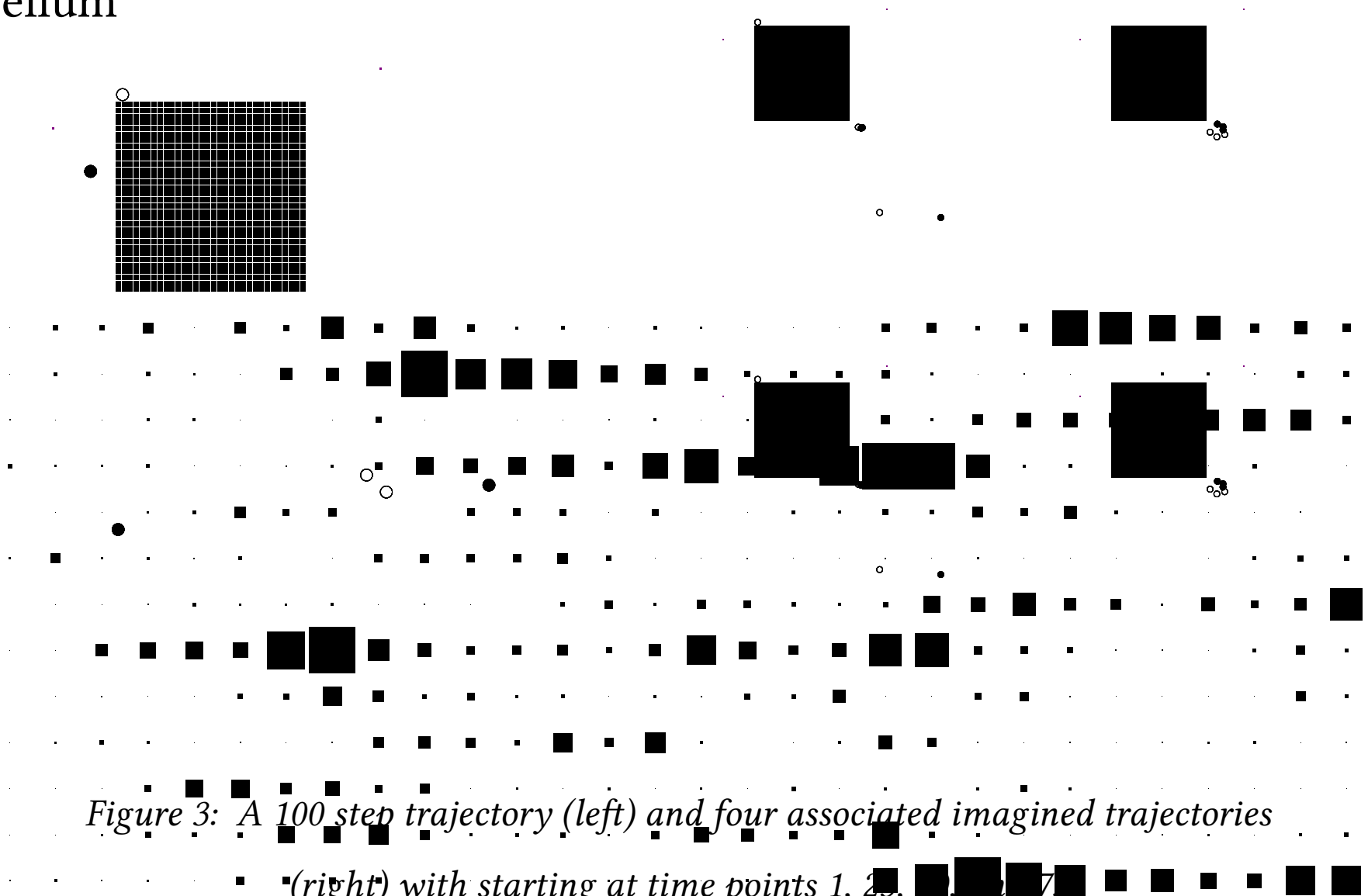


Figure 3: A 100 step trajectory (left) and four associated imagined trajectories (right) with starting at time points 1, 25, [illegible]

4.1 | Intel

- Recall distance was *increasing* in previous map
- Map dictates convergence vs. divergence?
- Relationship between masks and dynamics

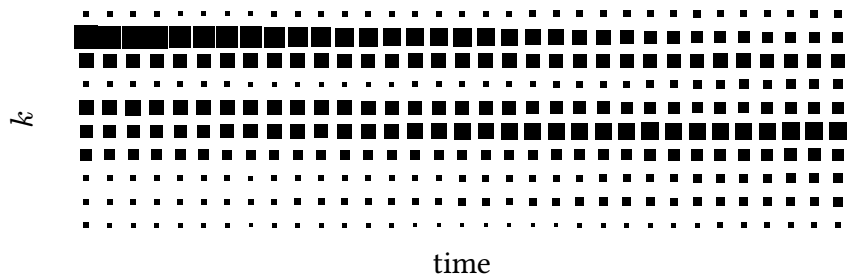
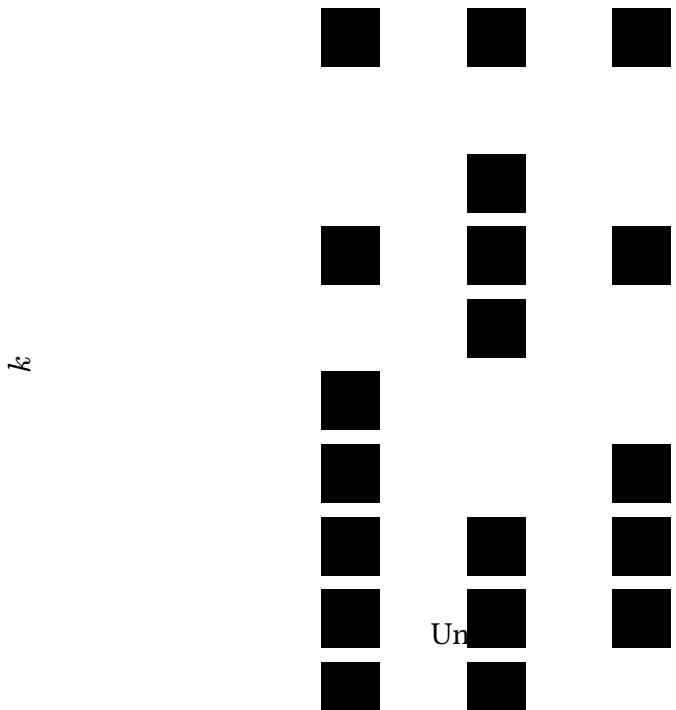
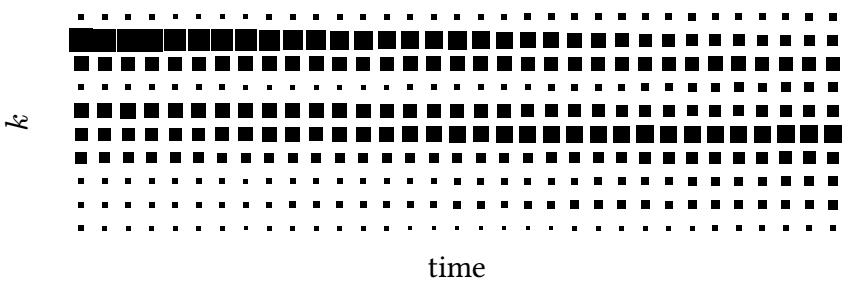


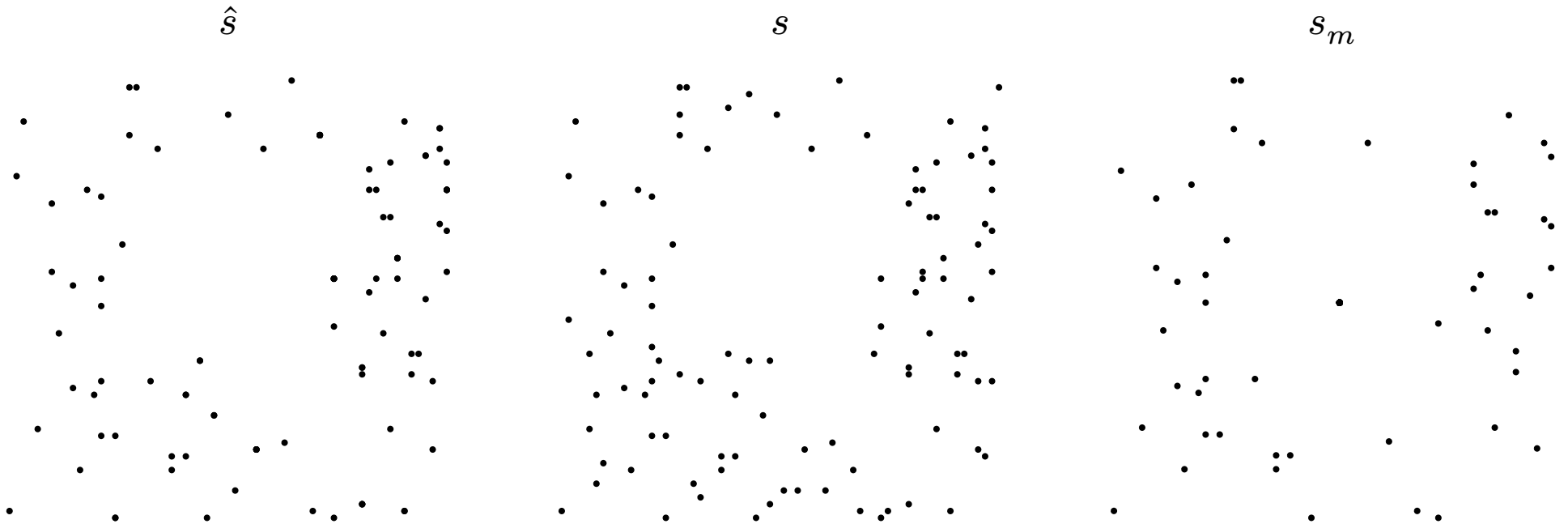
Figure 5: Distance between $\hat{s}_{i_k}$ and s_i over time

4.1 | Intel





4.1 | Intel



$\mathrm{CD}(\hat{s}, s) = 102$ $\mathrm{CD}(s, s_m) = 686$ $\mathrm{CD}(\hat{s}, s_m) = 712$

Table 1: TLDR: It works! $\hat{s}$, s, and s_m and associated Chamfer Distances. Note s is much closer to the estimate $\hat{s}$ than the masked s_m, showing that the LMM's processing of intel i is helpful.

4.2 | `detel_fn(intel_fn(s))`

- Using `gamma` (`jax` native and easy fine tuning)
- As per Figure 7 we:
  1. We generate language intel i_t from state s_t
  2. Mask away some (maybe all) of state (s_{mt})
  3. Decode i_t and s_{mt} to get estimate $s_{\hat{t}}$
- See Appendix A for intel string templates
- Status: did `intel_fn` and doing `detel_fn`

Function IntelFunction(s_t)
    Generate mask for units not in sight
    Generate i_t from s_t (could be lies)
    Hide parts of s_t using mask to produce s_{mt}
    return i_t, s_{mt}
end
Function DetelFunction(i_t, s_{mt})
    Create prompt requesting indices to update
    Use model to interpret i_t and s_{mt}
    Update s_{mt} with interpreted values
    return updated state estimate $\hat{s}_t$
end
$\hat{s}_t$) = DetelFunction(IntelFunction(s_t))

Figure 7: Pseudo code

5 | MIII

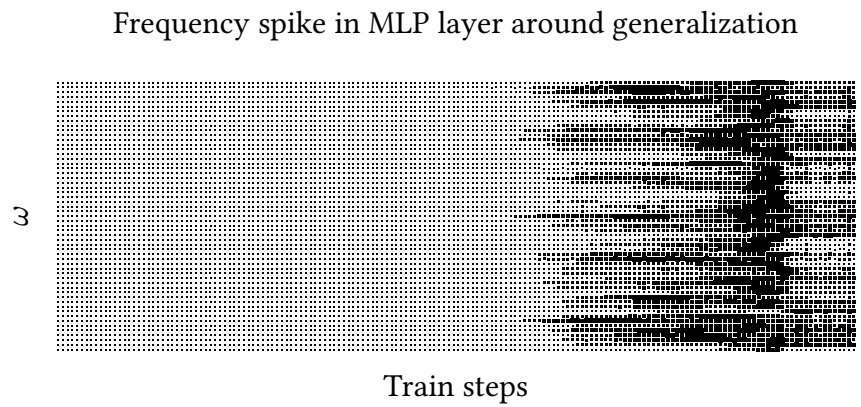


Figure 8: The spike in active frequencies during generalization indicate the presence of a non-generalizing and non-overfitting gradient component

- Grads have leaning and memory comps [1]
- Figure 8 Indicate a third, support-wheel comp
- Goal: publish in ICLR (better establish comp?)
- Now: chaning to better show spike across runs

6 | AIGS

MCTS

- Connect 4 pettingZoo [2]
- Implement MCTS
- Tweak params and compete

DRL

- Get unity ml-agent to run
- pick game. Use PPO.
- play against

QD

- implement map elite
- generate dataset of levels
- Play lebel with drl bot

Index of Sources

[1] J. Lee, B. G. Kang, K. Kim, and K. M. Lee, “Grokfast: Accelerated Grokking by Amplifying Slow Gradients,” no. arXiv:2405.20233. Jun. 2024.

[2] J. Terry *et al.*, “Pettingzoo: Gym for Multi-Agent Reinforcement Learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 15032–15043, 2021.

A | Intel templates

> "Breaking news from the battlefield: Allied forces report enemy combatant spotted at {pos} with approximately {hp} health remaining."

> "Hey, did you hear? My cousin saw someone lurking around {pos} yesterday. They looked pretty beat up, maybe only {hp} health left. Be careful out there."

> "URGENT DISPATCH: Target acquired at coordinates {pos}. Visual assessment indicates {hp} vitality points. Proceed with caution."

> "Journal Entry, Day 47: Today I encountered a strange figure at {pos}. They appeared wounded, perhaps {hp} strength remaining.."

> "According to reliable sources, an individual was recently sighted at {pos} in poor condition, estimated at {hp} health. Local authorities knows."

> "Overheard at the tavern: 'I'm telling you, I saw them clear as day at {pos}! Could barely stand, maybe {hp} health at most. Something's not right.'"

> "Scout's Log: Entity detected at position {pos}. Current status: {hp} hit points. Monitoring situation closely."

> "My grandmother always said to watch out for strangers at {pos}. Well, I just saw one there, and they only had about {hp} health by the looks of it."

> "MEDICAL REPORT: Patient last seen at location {pos} with critical injuries. Estimated {hp} health remaining. Immediate assistance required."

> "Text message received: 'omg just saw someone at {pos}!! they look hurt bad, maybe like {hp} health?? should we call someone???'"