

A SEMANTIC SIGNAL PROCESSING FRAMEWORK

Noah Syrkis

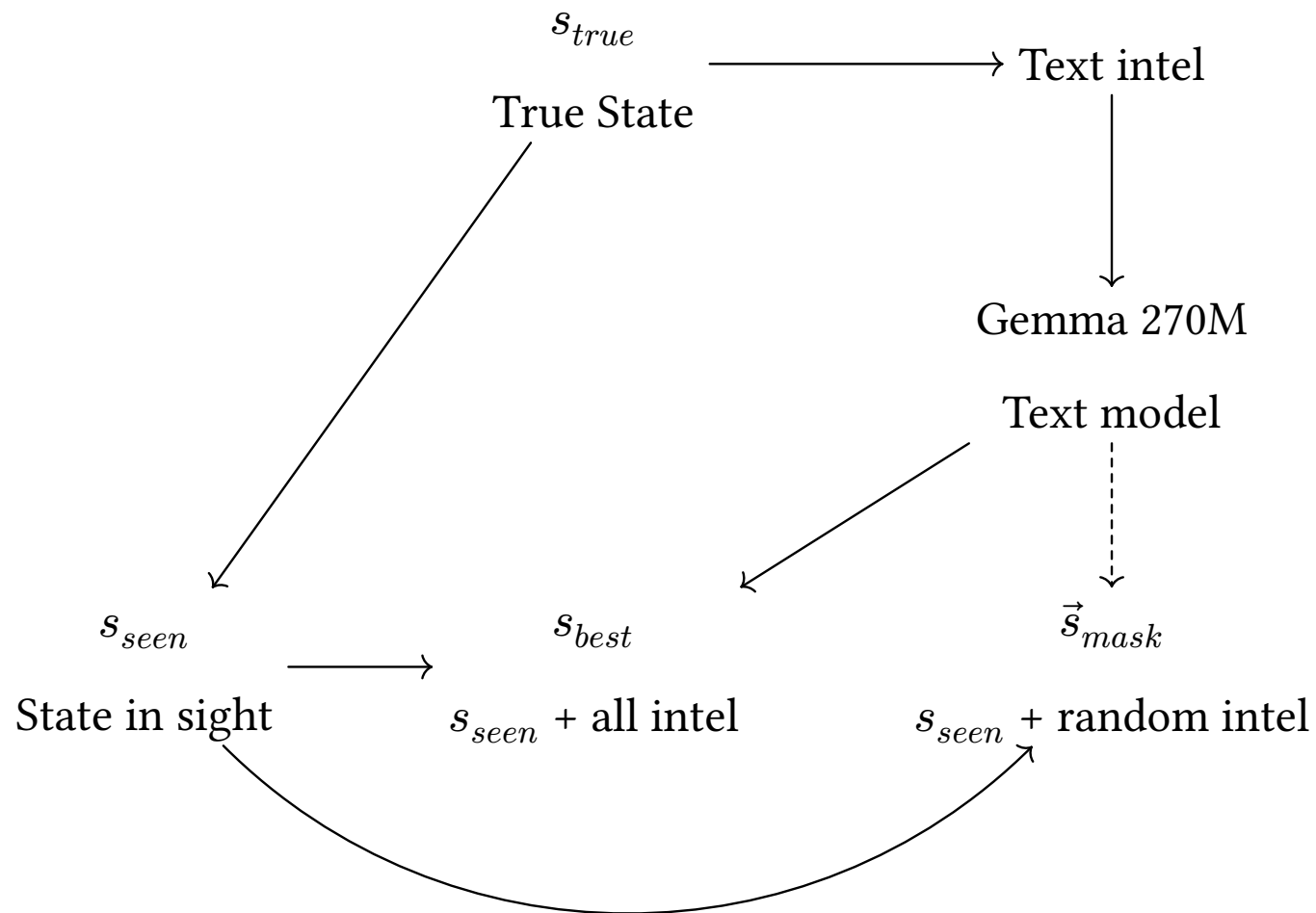
September 6, 2025

1 | Introduction

2 | Architecture

3 | Evaluation

4 | Future



- ▶ New RNG dimension for each of k sim sims
- ▶ Quantization of state (Figure 2) during play
- ▶ Today: improve masking (uniform in space)

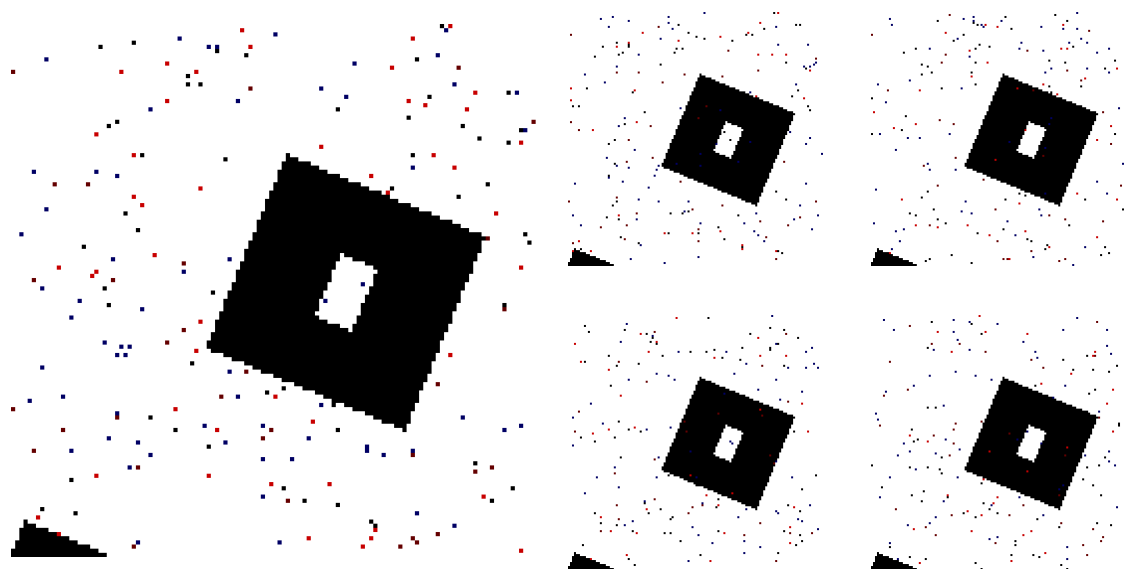


Figure 2: A single base simulation \vec{s}_t (left) and four imagined simulations $\vec{\bar{s}}_t$ (right)

1 | Introduction

Nebellum is a *semantic* signal processing framework. It leverages a multimodal large language model, Gemma [1], to decode pieces of intel, and a vectorized war game, Parabellum, to assign *importance* to those pieces.

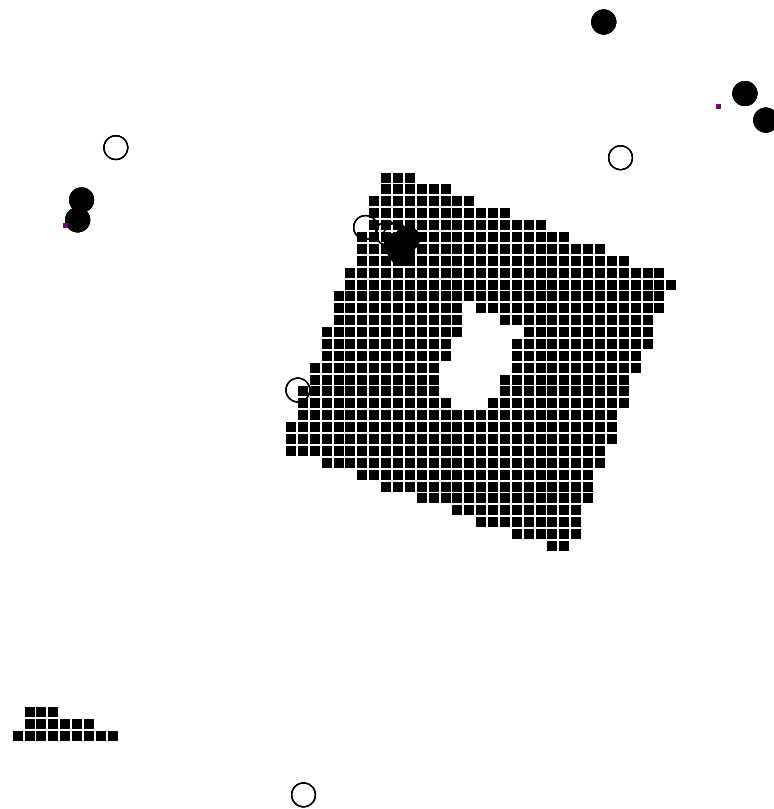


Figure 3: *Parabellum simulation of Colosseo*

1.1 | Notation

This is simulation heavy project. We have true game states, various kinds of estimated game states, as well as simulated game states

s_t	A true game state at time t
s'_t	Simulated game state based on s_t
\hat{s}_t	An intel subset based <i>estimate</i> of s_t
\hat{s}'_t	Simulated game state based on s'_t
\bar{s}_t	Estimte of s_t based on all intel
\bar{s}'_t	Simulated state based on \bar{s}_t

2 | Architecture

- ▶ Low level behavior is controlled by assigning behavior trees b and target positions
- ▶ Behavior trees map observations (info on units in sight range) to actions (move or shoot vector)
- ▶ Unit behavior (and target) is assigned by evaluating plan p at time t
- ▶ Plan evaluation happens m (evenly spaced) times throughout an n step episode

2 | Architecture

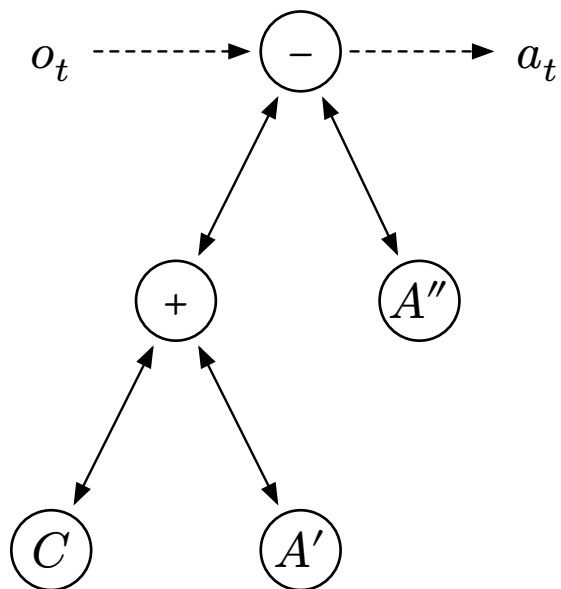


Figure 4: Behavior tree receiving a unit observation

o_t and returning a unit action a_t

- ▶ Leafs are conditions (C) or actions (A)
- ▶ Non-leafs are sequences (+) or fallbacks (-)
- ▶ They require all (+) or one (-) child success
- ▶ In Figure 4, if C succeeds A' runs else A'' runs
- ▶ “if enemy is in reach shoot enemy else move up”

2 | Architecture

- ▶ Plan evaluation maps state s_t to behavior b_t
- ▶ Figure 5 shows a graph with nodes that contain unit group g , target t and behavior b
- ▶ Targets can be positional (g shall go near t) ...
- ▶ ... or adversarial (g shall kill enemies near t)

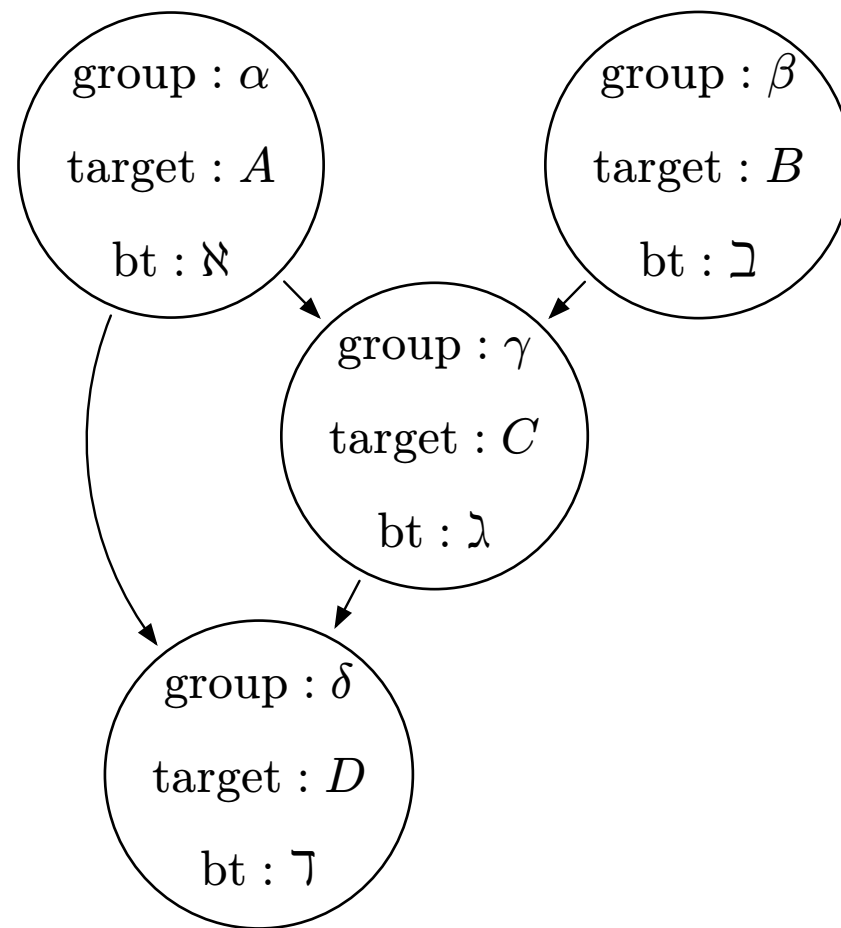


Figure 5: Example of plan data structure

2 | Architecture

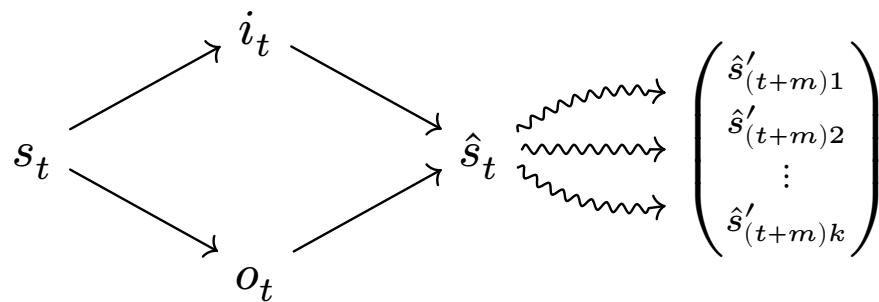


Figure 6: At time t we have state s_t , intel i_t and observation o_t . These are combined into \hat{s}_t , the basis for k different m step trajectories

Each of the k plan evaluations consists of:

1. Analysis: estimating the state \hat{s}_t by combining unit observation o_t with intel i_t
2. Simulation: setting k *imagined* simulations in motion based on \hat{s}_t

2.1 | Analysis

- ▶ Intel i_t is generated on all units in s_t encoding position and health in natural language
- ▶ Each piece of intel is fed to Gemma for analysis and combination into state estimate \hat{s}_t
- ▶ For each team, k intel subsets are made from a random masking of enemy units out of sight
- ▶ For each intel subset $i_{tj} : j \in [1, k]$ state \hat{s}_{tj} is made by masking info in \hat{s}_t from intel *not* in i_{tj}

{pos} !! look hurt, maybe {hp} health?'
⋮
a stranger at {pos}. health {hp}?

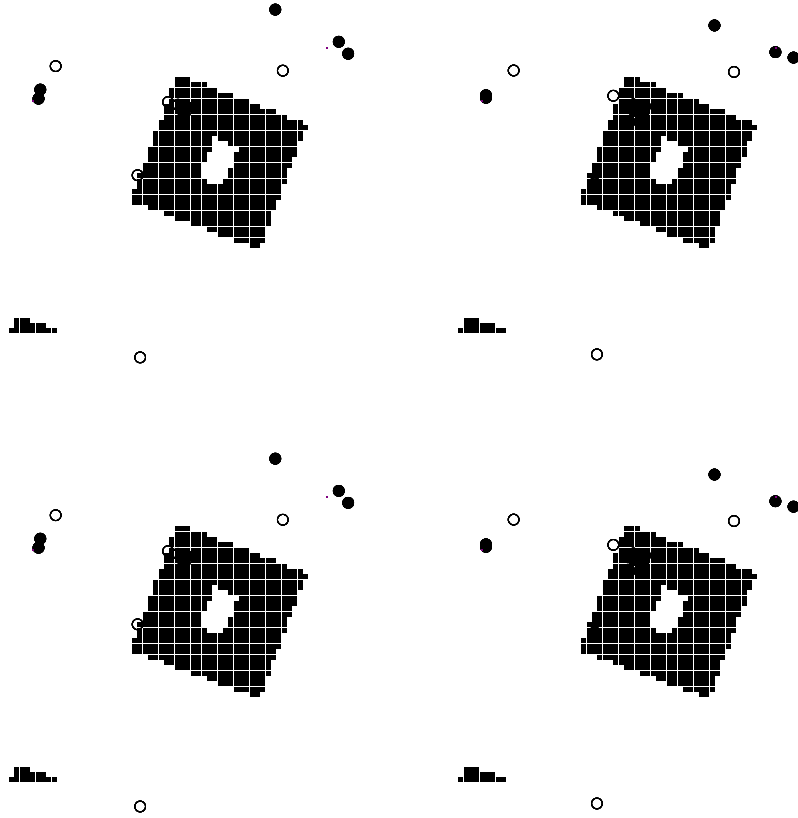


(x, y)	01
⋮	⋮
(x, y)	01



$(\hat{s}_{t1} \ \hat{s}_{t2} \ \dots \ \hat{s}_{tk})$

2.2 | Simulation



- ▶ The plan p is evaluated for each of the k states $\hat{s}_{tj} : j \in [1, k]$ yielding the k behaviors b_j
- ▶ From s_t or \hat{s}_{tj} , k trajectories (one for each behavior b_j) of length $\lfloor \frac{n}{m} \rfloor$ are run and recorded

Figure 7: Simulated futures based on would be behaviors based on different pieces of intel

2.2 | Simulation

- ▶ Using s_t as simulation basis we can assign importance to different aspects of the intel i_{tj}
- ▶ Using \hat{s}_{tj} we can assign importance to different aspects of the current state s_t

3 | Evaluation

- ▶ As expected the delta starts small
- ▶ Divergence occurs under some conditions
- ▶ For a subset of these, it converges again

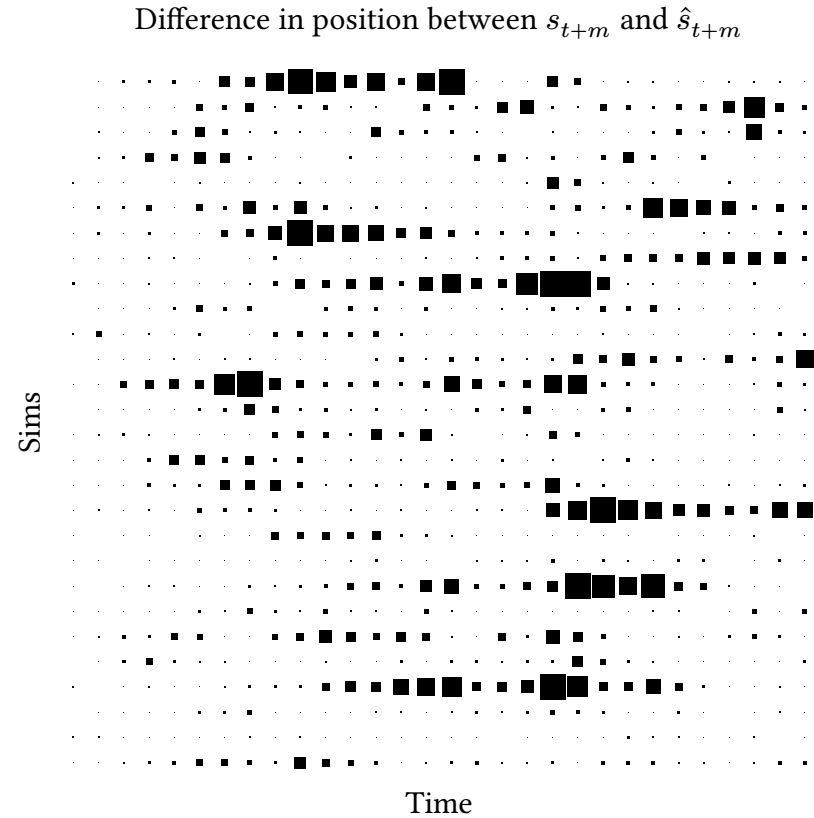
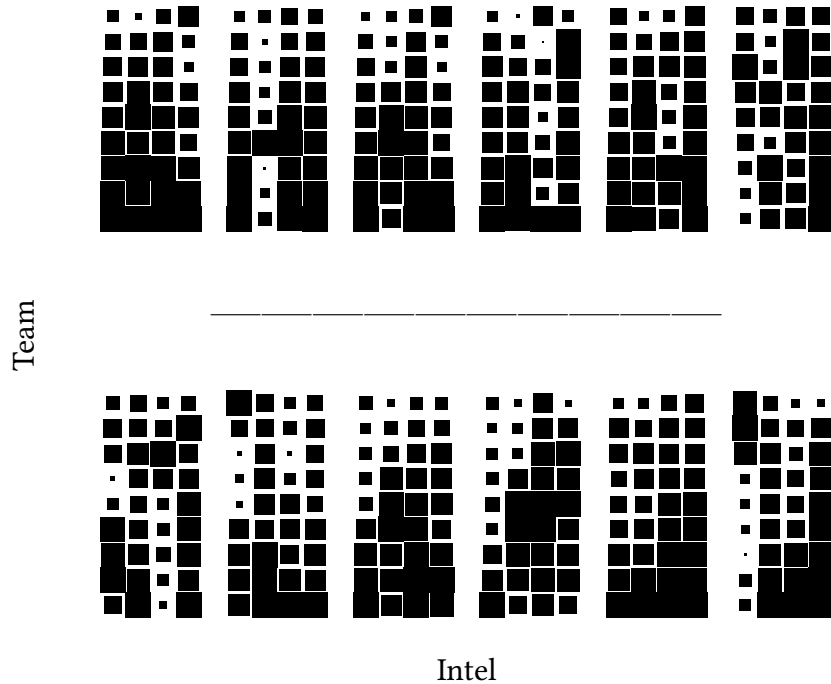


Figure 8: Divergence between true state s_t and simulated states. Simulations are based on s_t (not \hat{s}_t . They thus differ in behavior and random seed.

3 | Evaluation



- ▶ In subplots rows is epoch and cols is time
- ▶ We see that there *is* a pattern
- ▶ Sometimes somethings predict the future ...
- ▶ ... other times other things predict the future

*Figure 10: Linear regression β values of importance
of different pieces of intel*

4 | Future

- ▶ ☒ Compute importance score
- ▶ ☐ Highlight which intel is important
- ▶ ☐ Add importance overlay to location on top of map (nice UI)
- ▶ ☐ Multiple trajectories s' per intel subset (for robustness)
- ▶ ☐ Hide true enemy plan, inferring it from o_t and i_t
- ▶ ☐ Non visible enemy units could be sampled uniformly?

Index of Sources

- [1] G. Team *et al.*, “Gemma 3 Technical Report,” no. arXiv:2503.19786. arXiv, Mar. 2025. doi: 10.48550/arXiv.2503.19786. **(2)**