Formula Collection

Emergent Intelligence and Semantic Pressure

Adrian Zander

20.05.2025

Contents

Formula Collection

1. Semantic Pressure (SP)

$$SP = \alpha H(T) + \beta S(I) + \gamma D(C) \tag{1}$$

Explanation: Semantic Pressure (SP) quantifies the complexity of a prompt for a language model.
Function:

- $H(T)$: Token entropy (uncertainty in word prediction)
- $S(I)$: Sentiment load (emotional intensity)
- $D(C)$: Context divergence (misalignment with prior context)
- α, β, γ: Weights (typically $\frac{1}{3}$ each)

2. Context-Adjusted Semantic Pressure

$$SP_{\text{adjusted}} = SP - \lambda_c R(C) \tag{2}$$

Explanation: Adjusts the SP score based on the coherence $R(C)$ of the context.
Function:

- $R(C) \in [0, 1]$: Coherence score
- λ_c: Retention weight (e.g., 0.2)

3. Statistical Threshold for SP

$$SP_{\text{thr}} = \mu + k\sigma \tag{3}$$

Explanation: Defines a threshold for high semantic pressure.
Function:

- μ: Mean SP for stable prompts
- σ: Standard deviation
- k: Confidence factor (e.g., $k = 2$ for 95%)

4. Adaptive Temperature Scaling

$$\tau(SP_{\text{adjusted}}) = \tau_0 \left(1 + \lambda_\tau \max(0, SP_{\text{adjusted}} - SP_{\text{thr}})\right) \tag{4}$$

Explanation: Dynamically adjusts the sampling temperature of the language model based on semantic pressure.
Function:

- τ_0: Base temperature (e.g., 0.7)
- λ_τ: Adjustment factor (e.g., 0.5)

5. Attractor Health Model (Adrian Thesis)

$$\frac{dx_i}{dt} = f_i(x_1, x_2, ..., x_n) + \epsilon_i(t) \tag{5}$$

$$\text{Health} := \lim_{t\to\infty} x(t) \in A \subset \mathbb{R}^n \tag{6}$$

Explanation: Describes the evolution of system health as a vector $x(t)$ in state space, subject to dynamics f_i and perturbations $\epsilon_i(t)$. Health is convergence to an attractor set A.

6. Decentralized Consensus (Adrian Thesis)

$$x_i(t+1) = \sum_{j\in N(i)} \alpha_{ij} x_j(t), \quad \sum_j \alpha_{ij} = 1 \tag{7}$$

$$\kappa(t) = \text{Var}(x_i(t)), \quad \lim_{t\to\infty} \kappa(t) = 0 \tag{8}$$

Explanation: Models how agents reach consensus via local averaging. Variance $\kappa(t)$ vanishes as consensus is achieved.

7. Resonant Semantic Pressure (RSA)

$$RSP = \alpha H(T) + \beta S(I) + \gamma D(C) + \delta\langle \phi(m), \vec{U}\rangle \tag{9}$$

Explanation: Extends SP by adding an ethical alignment term.
Function:

- $\langle \phi(m), \vec{U}\rangle$: Alignment of message intent with utility vector
- δ: Weight for ethical alignment

Summary Table

Formula	**Meaning**
$SP = \alpha H(T) + \beta S(I) + \gamma D(C)$	Semantic Pressure (prompt complexity)

$SP_{\text{adjusted}} = SP - \lambda_c R(C)$	Context-adjusted SP
$SP_{\text{thr}} = \mu + k\sigma$	Statistical SP threshold
$\tau(SP_{\text{adjusted}}) = \tau_0(1 + \lambda_\tau \max(0, SP_{\text{adjusted}} - SP_{\text{thr}}))$	Adaptive temperature scaling
$\frac{dx_i}{dt} = f_i(x_1, ..., x_n) + \epsilon_i(t)$	System health dynamics
$x_i(t+1) = \sum_{j \in N(i)} \alpha_{ij} x_j(t)$	Decentralized consensus update
$RSP = \alpha H(T) + \beta S(I) + \gamma D(C) + \delta \langle \phi(m), \vec{U} \rangle$	Resonant Semantic Pressure (RSA)

References

For further details and algorithmic examples, see the main chapters of the Zander Collection.

Appendix: Python Code Examples for Formula Collection

1. Semantic Pressure (SP)

```python
def sp(H, S, D, alpha=1/3, beta=1/3, gamma=1/3):
    return alpha * H + beta * S + gamma * D

# Example values
H = 0.9     # Token entropy
S = 0.7     # Sentiment load
D = 0.6     # Context divergence
SP = sp(H, S, D)
print(f"SP: {SP:.3f}")  # Output: SP: 0.733
```

Listing 1: Semantic Pressure (SP) Example

2. Context-Adjusted Semantic Pressure

```python
SP = 0.733
lambda_c = 0.2
R_C = 0.8    # Context coherence score
SP_adj = SP - lambda_c * R_C
print(f"SP_adjusted: {SP_adj:.3f}")  # Output: SP_adjusted: 0.573
```

Listing 2: Context-Adjusted SP Example

3. Statistical Threshold for SP

```python
mu = 0.3      # Mean SP for stable prompts
sigma = 0.1   # Standard deviation
k = 2         # Confidence factor (e.g., 2 for 95%)
SP_thr = mu + k * sigma
print(f"SP_thr: {SP_thr:.3f}")  # Output: SP_thr: 0.500
```

Listing 3: Statistical Threshold for SP Example

4. Adaptive Temperature Scaling

```python
SP_adj = 0.65
SP_thr = 0.5
tau_0 = 0.7
lambda_tau = 0.5
tau = tau_0 * (1 + lambda_tau * max(0, SP_adj - SP_thr))
print(f"Adaptive temperature: {tau:.3f}")  # Output: Adaptive
    temperature: 0.753
```

Listing 4: Adaptive Temperature Scaling Example

5. Attractor Health Model (Adrian Thesis)

```python
import numpy as np

def dxdt(x, f, eps):
    return f(x) + eps

# Example: Linear system
f = lambda x: -0.5 * x
eps = lambda t: 0.01 * np.random.randn()
x = 1.0
for t in range(100):
    x += dxdt(x, f, eps(t))
print(f"Final x: {x:.3f}")
```

Listing 5: Attractor Health Model Example

6. Decentralized Consensus (Adrian Thesis)

```python
import numpy as np

x = np.array([0.1, 0.5, 0.9])
A = np.array([[0.5, 0.25, 0.25],
              [0.25, 0.5, 0.25],
              [0.25, 0.25, 0.5]])
for _ in range(20):
    x = A @ x
print(f"Consensus: {x}")
```

Listing 6: Decentralized Consensus Example

7. Resonant Semantic Pressure (RSA)

```python
def rsp(H, S, D, ethical, alpha=0.25, beta=0.25, gamma=0.25,
    delta=0.25):
    return alpha*H + beta*S + gamma*D + delta*ethical

```

```python
# Example values
RSP = rsp(0.8, 0.6, 0.7, 0.9)
print(f"RSP: {RSP:.3f}")  # Output: RSP: 0.750
```

Listing 7: Resonant Semantic Pressure (RSA) Example