* [Bio-plausible digital implementation of a reward modulated STDP synapse | Neural Computing and Applications](https://link.springer.com/article/10.1007/s00521-022-07220-6#Abs1) (Quintana 2022)
  + <https://gist.github.com/ferqui/a96a7120c9eb61747987b8b021dce7fc>
  + Uses FPGA, Brian2 (want to use for neuromorphic + PyTorch)
  + How do they define the reward signal? They manually set it >:(
    - Goal: Demonstrate that R-STDP can learn adaptive behavior in a real-world task.
    - Experimental Setup:
      * Robot: Equipped with 6 ultrasonic sensors (inputs) and 2 motor neurons (outputs).
      * SNN Architecture:
        + 6 input neurons (I&F model) connected directly to 2 output neurons (no hidden layers).
        + Synapses used biexponential current dynamics (decay time = 3 ms).
      * Learning Rule:
        + R-STDP modulated by dopamine, which was generated based on deviation from a Braitenberg controller’s output.
        + Dopamine acted as a reward prediction error:

Positive reward if the SNN’s output matched the Braitenberg solution.

Negative/no reward otherwise.

* + - Training:
      * Prerecorded sensor-motor data (from Braitenberg algorithm) was used for supervised training.
      * The network updated weights every 64 ms.
    - Results:
      * After 6 seconds of training, the SNN achieved 95% accuracy in choosing the correct turning direction (Figure 10).
      * Weight distribution (Figure 11) showed:
        + Left motor neuron strongly connected to left-side sensors.
        + Right motor neuron strongly connected to right-side sensors.
      * Key advantage: The SNN learned without a hidden layer, reducing computational complexity.
  + Why did they manually inject DA at t=20ms? No real environmentally based reward function
* [Supervised Learning in SNN via Reward-Modulated Spike-Timing-Dependent Plasticity for a Target Reaching Vehicle](https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2019.00018/full) (Bing et al., 2019)
  + The reward signal in this implementation of Reward-Modulated STDP (R-STDP) is supervised, meaning it is based on a pre-generated dataset rather than real-time environmental feedback.
  + Not real RL
* [Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling | Cerebral Cortex | Oxford Academic](https://academic.oup.com/cercor/article-abstract/17/10/2443/314939) (Izhekevich 2007)
  + Waiting to hear back about access
* [Reinforcement Learning in a Neurally Controlled Robot Using Dopamine Modulated STDP](https://arxiv.org/pdf/1502.06096)
  + A grad thesis, super well/simply/concisely written
  + https://arxiv.org/pdf/1502.06096
* [A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0893608019302680) (Hao 2020)
  + Input Presentation:
    - An input pattern (e.g., **MNIST** digit) is encoded as Poisson spike trains.
    - Spikes propagate through the SNN, generating output spikes in the final ("supervised") layer.
  + Teacher Signal Generation:
    - If the output neuron corresponding to the correct label fires, DA is released.
    - Example: For digit "3", only the "3" output neuron’s firing triggers DA.
  + Weight Updates via sym-STDP + DA:
    - sym-STDP
      * Unlike classic STDP (which strengthens synapses only if pre-spikes precede post-spikes), sym-STDP strengthens synapses regardless of spike order within a short time window.
    - Eligible synapses (those with recent pre/post spikes) are potentiated if DA is present.
    - Synaptic scaling and dynamic thresholds ensure stability.
  + No DA = No Learning:
    - If the wrong output neuron fires, no DA is released, and no plasticity occurs.
  + As the learning rules were bio-plausible and based purely on local spike events, our model could be easily applied to neuromorphic hardware for online training and may be helpful for understanding SL information processing at the synaptic level in biological neural systems
* [End to End Learning of Spiking Neural Network based on R-STDP for a Lane Keeping Vehicle](https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8460482) (Bing et al., 2018)
  + the lane information is encoded by the event data from a neuromorphic vision sensor
  + “The reward given to the SNN is defined for each motor individually as a linear function of the lane center distance”
  + “However, practical robotic implementations based on R-STDP are rarely found due to its complexity in feeding sensor data into SNNs, constructing and assigning the reward to neurons, and training the SNNs”
  + “For future work, the R-STDP controller is build as first step towards more sophisticated algorithms with real reinforcement learning capabilities. Currently research has not incorporated reward prediction errors yet, even though this phenomenon was observed in the brain. Therefore, such networks based on R-STDP should be also implemented using deep architectures in the future”
* [R-STDP Spiking Neural Network Architecture for Motion Control on a Changing Friction Joint Robotic Arm](https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2022.904017/full) (Juarez-Lora et al., 2022)
  + Neuromorphic computing based
  + “This study proposes a new neural architecture using the spike-time-dependent plasticity learning method and step-forward encoding algorithm for a self tuning neural control of motion in a joint robotic arm subjected to dynamic modifications.”
  + “This article proposes an SNN architecture that learns how to reconstruct an input signal.”
  + Reward function is a function of acceleration error
* [Fine-tuning Deep Reinforcement Learning Policies with r-STDP for Domain Adaptation](https://dl.acm.org/doi/pdf/10.1145/3546790.3546804)(Akl et al., 2022)
  + we train SNNs with backpropagation using surrogate gradients and the (Deep Q-Network) DQN algorithm to solve two classical control reinforcement learning tasks
  + To compensate for the drop in performance, we apply the biologically plausible reward-modulated spike timing dependent plasticity (r-STDP) learning rule. Our results show that r-STDP can be successfully utilized to restore the network’s ability to solve the task
  + Since such three-factor learning rules are available on neuromorphic devices [8, 25], we believe that combining backpropagation-based offline training with r-STDP online fine-tuning can provide a promising research direction for neuromorphic sim-to-real transfer, and could potentially open the door for more neuromorphic robotic applications.

Questions:

* How do they define reward signals?
  + How to create external reward signal aka dopamine increase = f(external factors)
    - If the robot avoids an obstacle, dopamine is released.
  + Imagine using vision language action (VLA) model to define a reward function (overkill?)
  + Reward prediction error? Is this even applicable to R-STDP?

Gaps in research

* Defining reward signal function
  + Making more generalizable reward function rule
* Tasks in which R-STDP is used for
  + MNIST (already done by sym-STDP paper, but could use R-STDP rules from other papers)
  + Reinforcement Learning Robotics
* Defining R-STDP for neuromorphic computing
  + Could recreate any one of these
* Honestly could create a RL for neuromorphic computing in general for our lab (will be so hard)
* Global vs local reward

**Basal ganglia R-STDP models (use RPE)**

* [Learning to Select Actions with Spiking Neurons in the Basal Ganglia - PMC](https://pmc.ncbi.nlm.nih.gov/articles/PMC3269066/#sec5)
* [A spiking Basal Ganglia model of synchrony, exploration and decision making](https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2015.00191/full#h3)