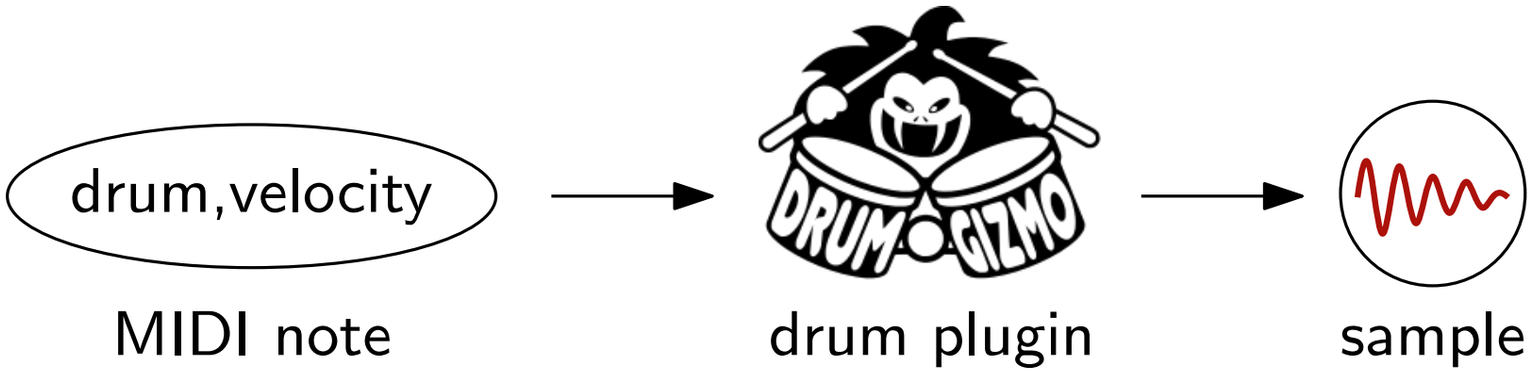
On Choosing Best Samples for Virtual Drums

André Nusser & Bent Bisballe Nyeng

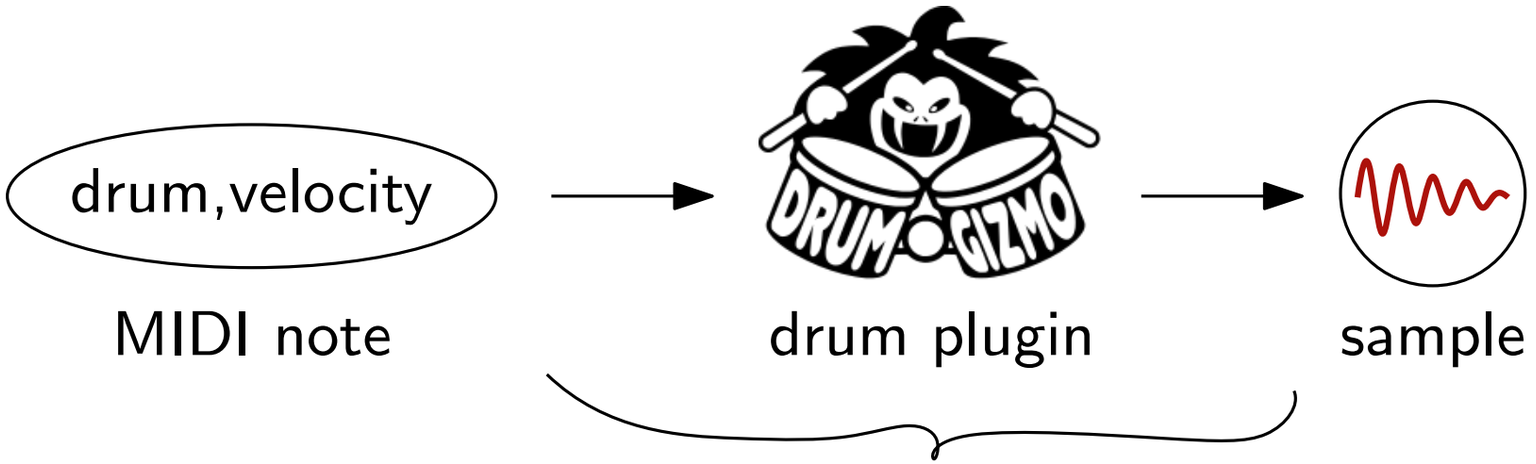
Motivation

How to select samples?



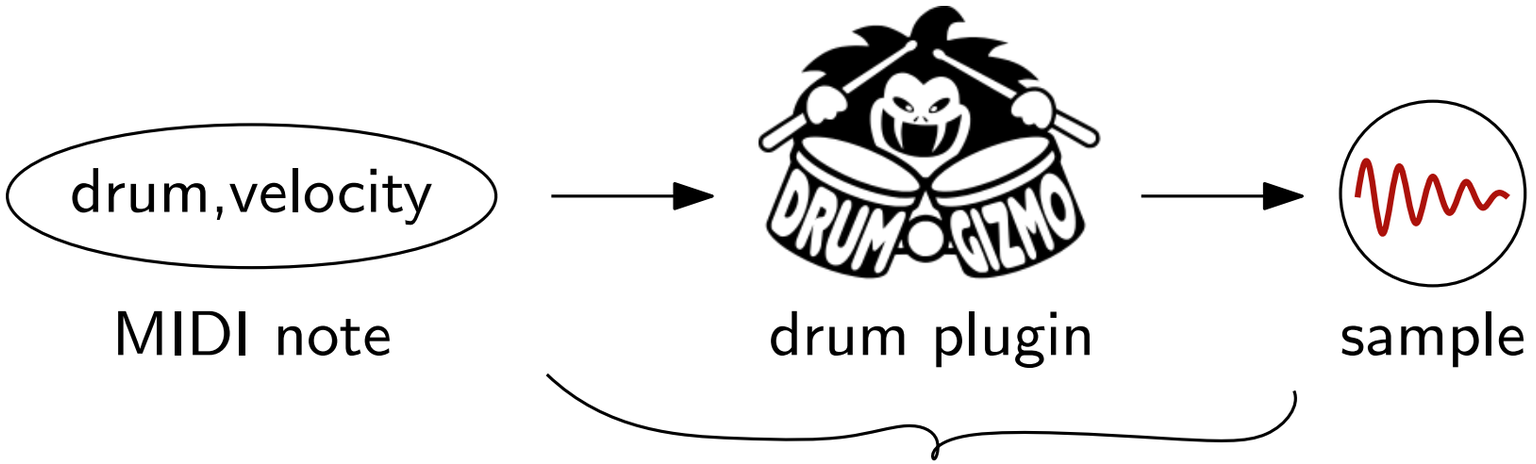
Motivation

How to select samples?

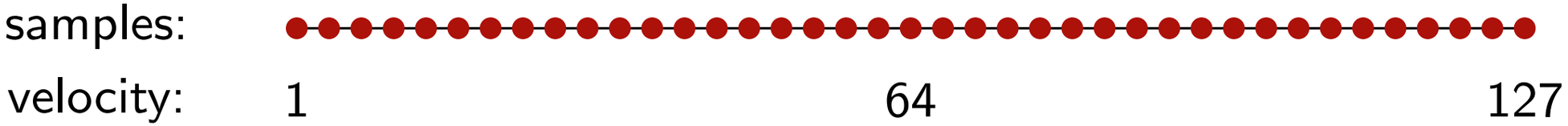


Motivation

How to select samples?

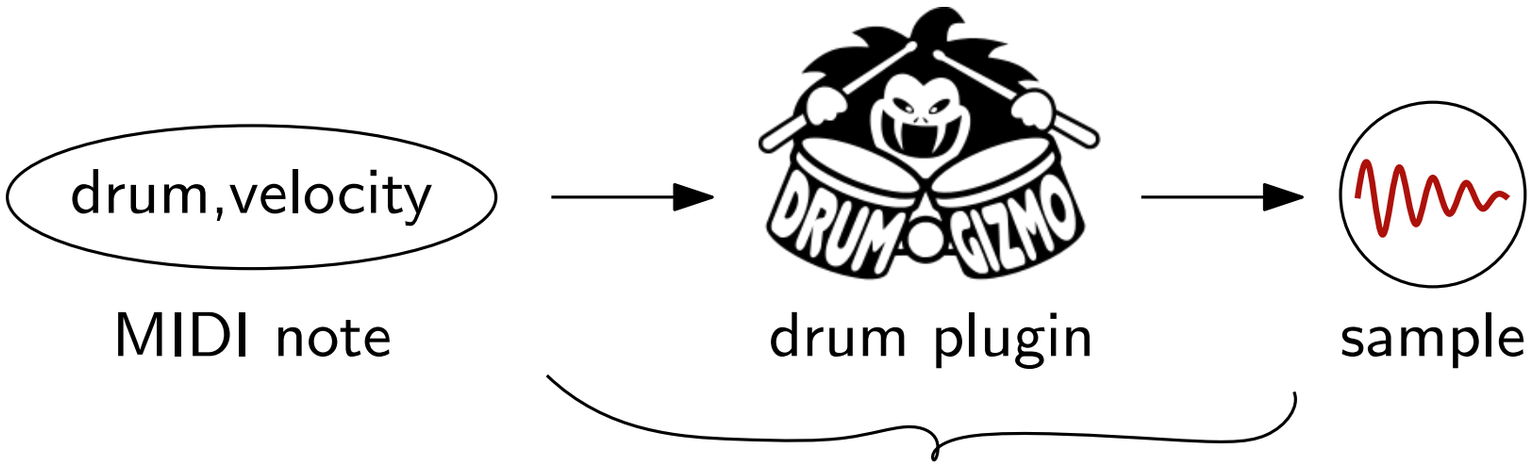


Naive Solution:



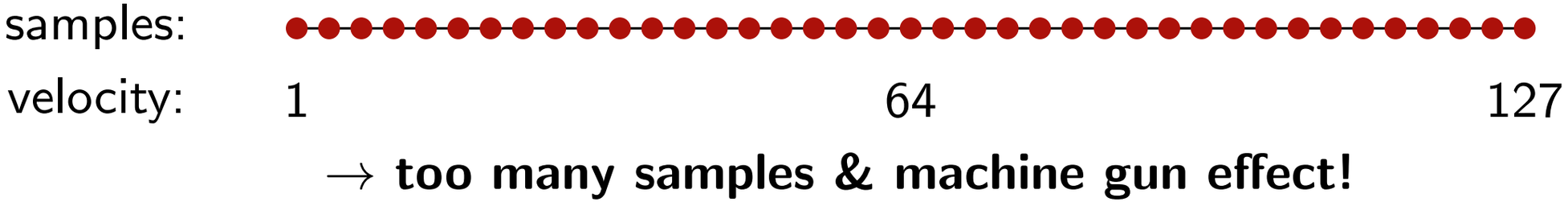
Motivation

How to select samples?



what happens here??

Naive Solution:

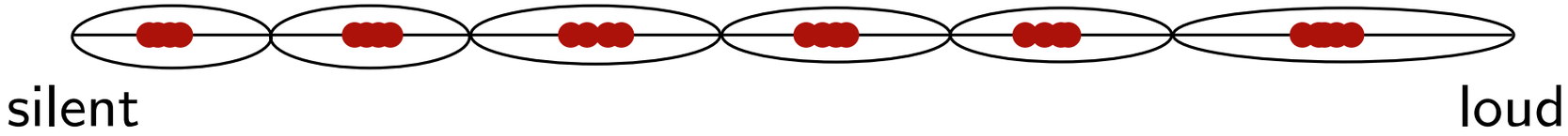


$\rightarrow$ **too many samples & machine gun effect!**

Motivation

Other Selection Methods

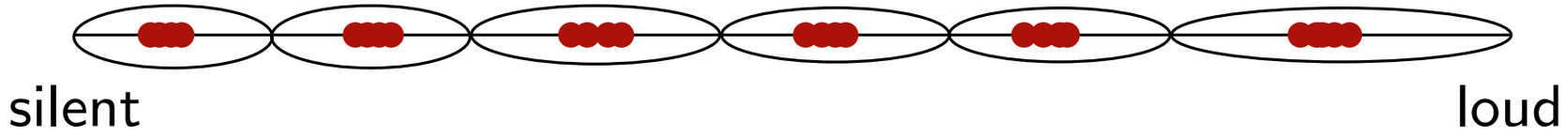
Round Robin:



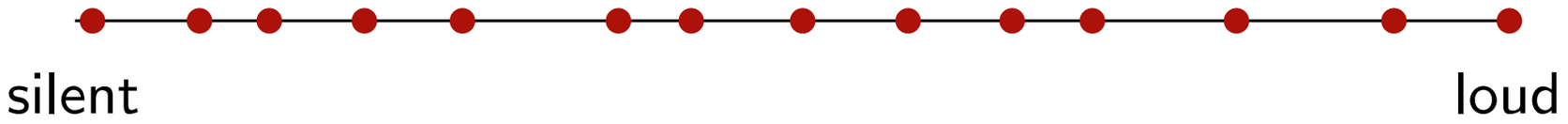
Motivation

Other Selection Methods

Round Robin:



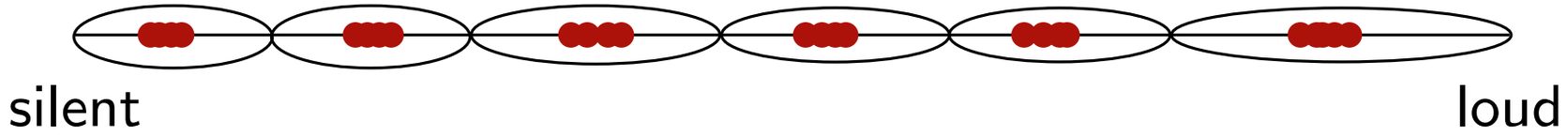
Continuous Velocities:



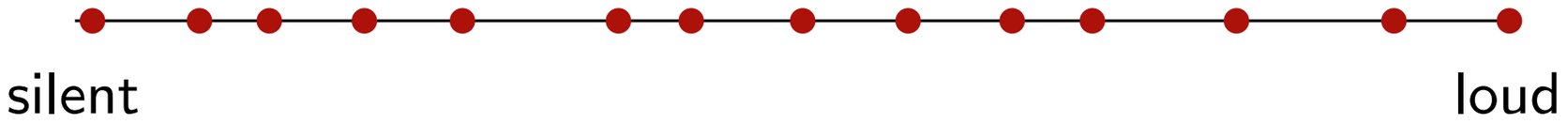
Motivation

Other Selection Methods

Round Robin:



Continuous Velocities:



→ **Remainder: How to select samples in this setting?**

Motivation

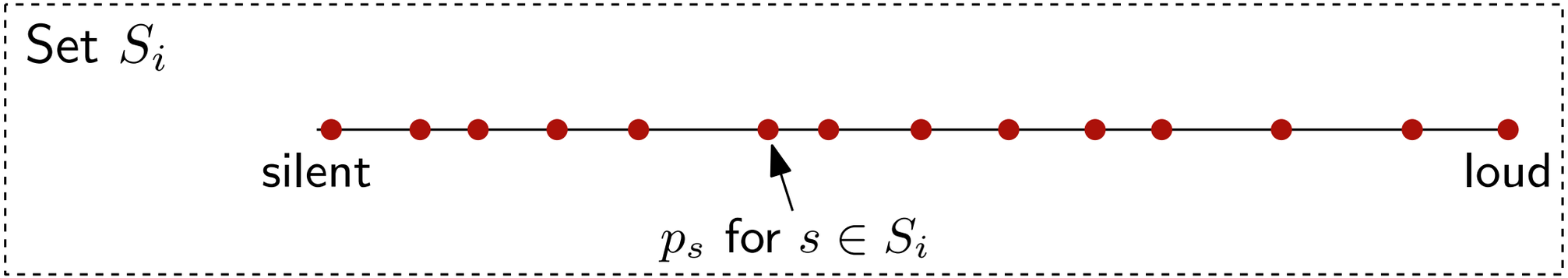
Why continuous velocities?

- sampling a drum kit is easier
- no staircase effect
- no scaling samples
- more flexible

Setting

Data:

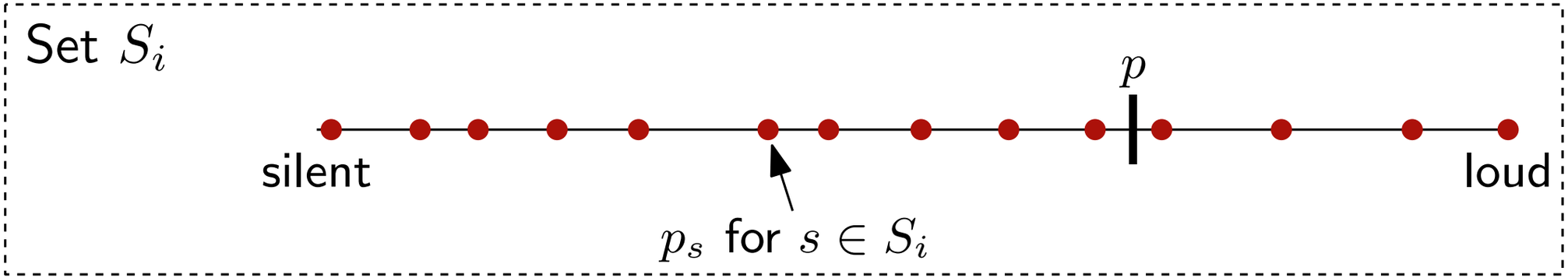
Instrument $i \in I$:



Setting

Data:

Instrument $i \in I$:



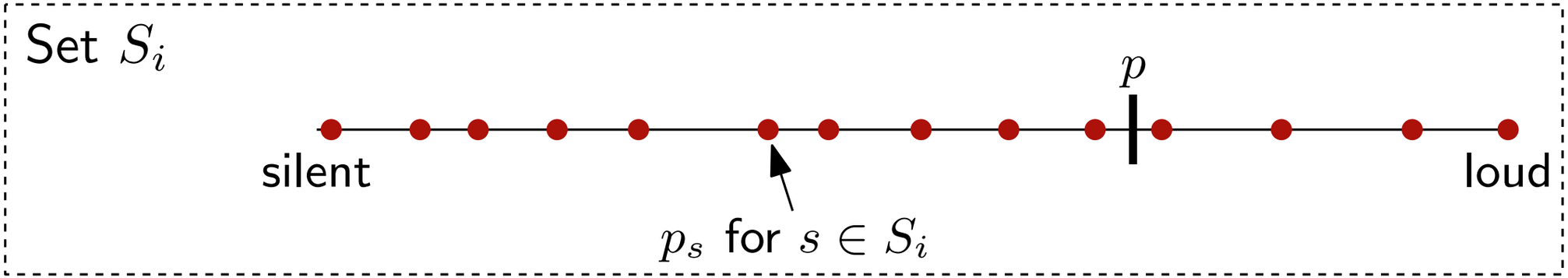
Requests: $(i, p) \in I \times \mathbb{R}^+$

Answer: best sample from S_i for the power value p

Setting

Data:

Instrument $i \in I$:



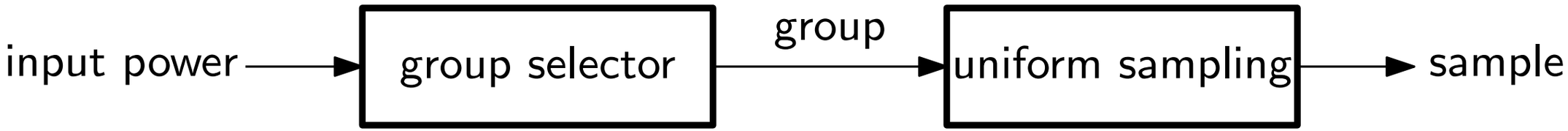
Requests: $(i, p) \in I \times \mathbb{R}^+$

Answer: best sample from S_i for the power value p

$\rightarrow$ **What is the best sample?**

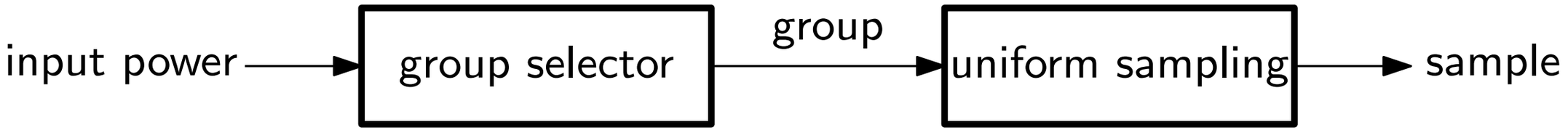
Previous Approaches in DrumGizmo

Approach I: Velocity Groups



Previous Approaches in DrumGizmo

Approach I: Velocity Groups

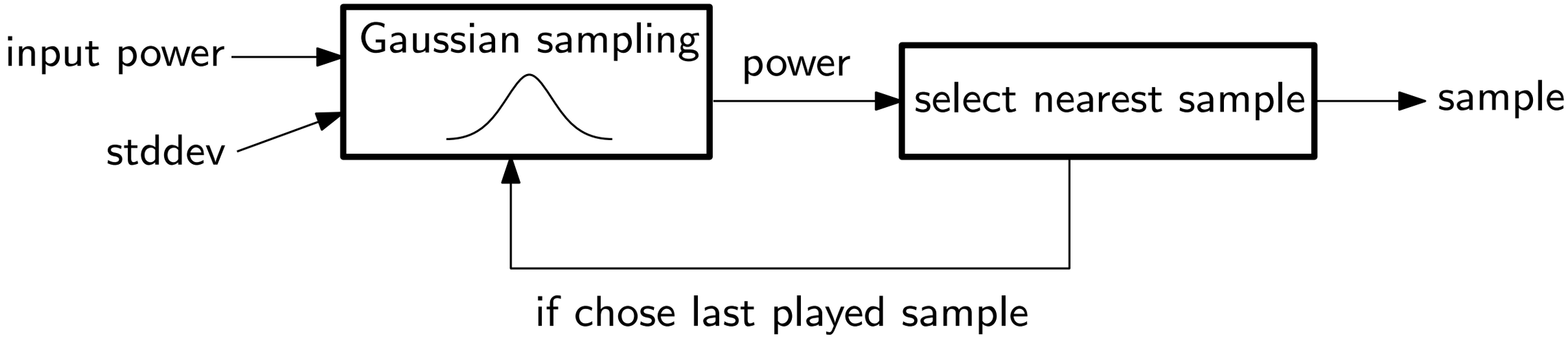


Issues:

- Staircase effect
- More work to create a drum kit

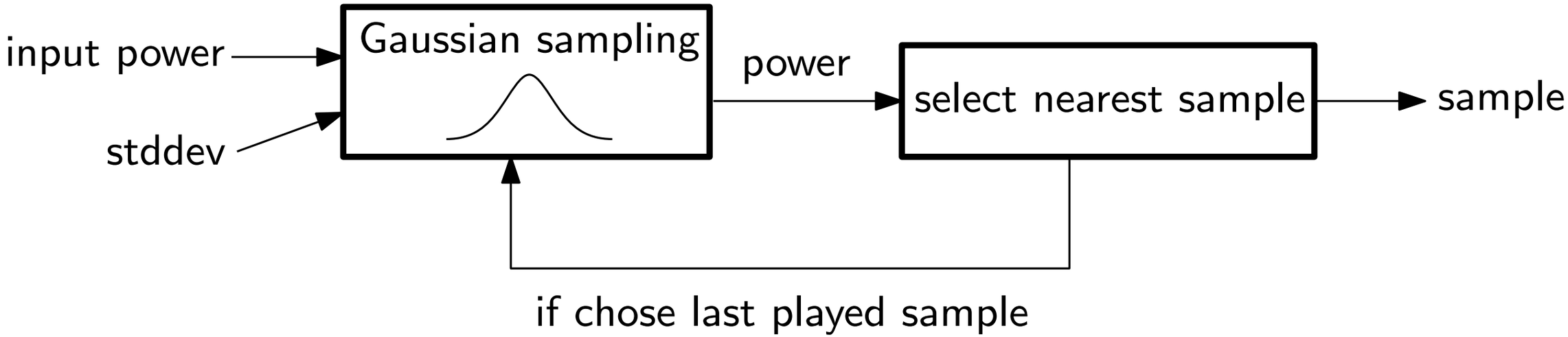
Previous Approaches in DrumGizmo

Approach II: Sampling from Normal Distribution



Previous Approaches in DrumGizmo

Approach II: Sampling from Normal Distribution



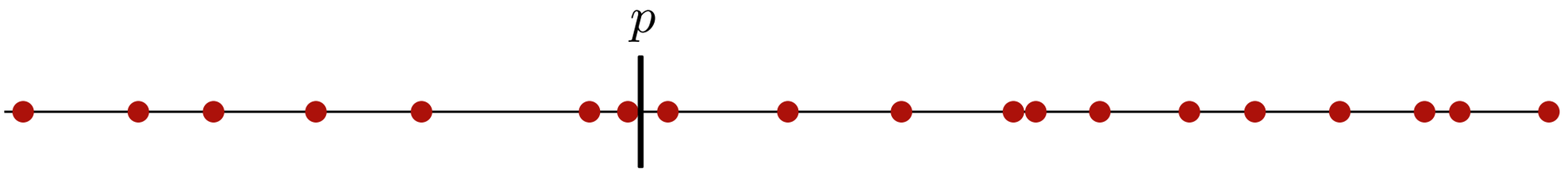
Issues:

- Possible machine gun effect
- Does not use full potential of data set:
- No notion of time

Requirements of Sample Selection Algorithm

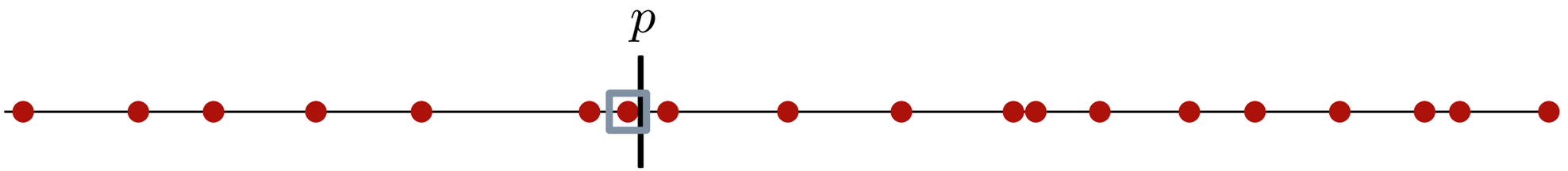
1) Closeness

Requirements of Sample Selection Algorithm



1) Closeness

Requirements of Sample Selection Algorithm



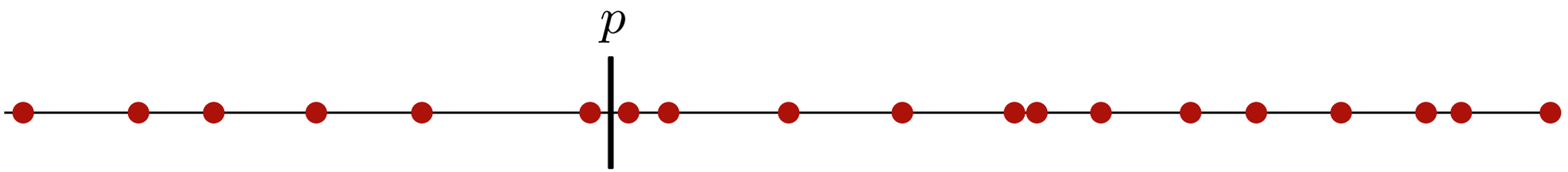
1) Closeness

Requirements of Sample Selection Algorithm

1) Closeness

2) Diversity

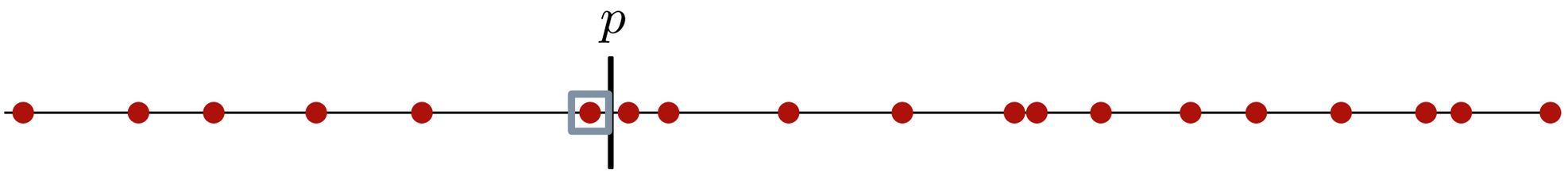
Requirements of Sample Selection Algorithm



1) Closeness

2) Diversity

Requirements of Sample Selection Algorithm



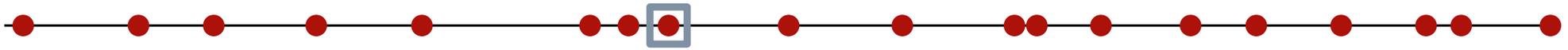
1) Closeness

2) Diversity

Requirements of Sample Selection Algorithm

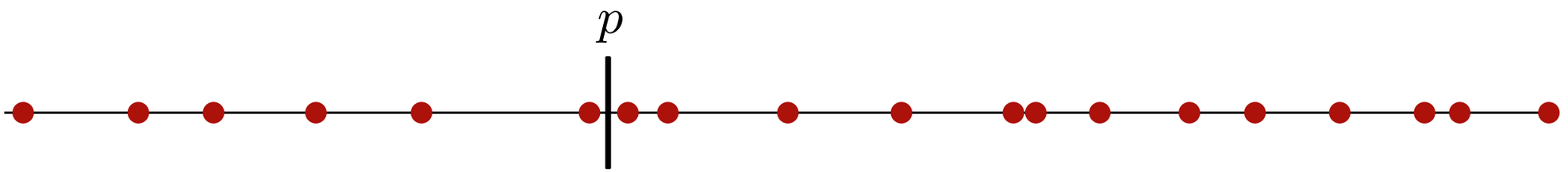
1) Closeness
2) Diversity
3) Randomization

Requirements of Sample Selection Algorithm



1) Closeness
2) Diversity
3) Randomization

Requirements of Sample Selection Algorithm

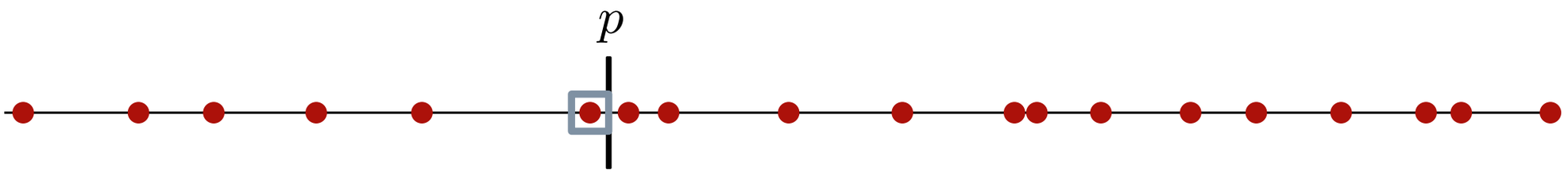


1) Closeness

2) Diversity

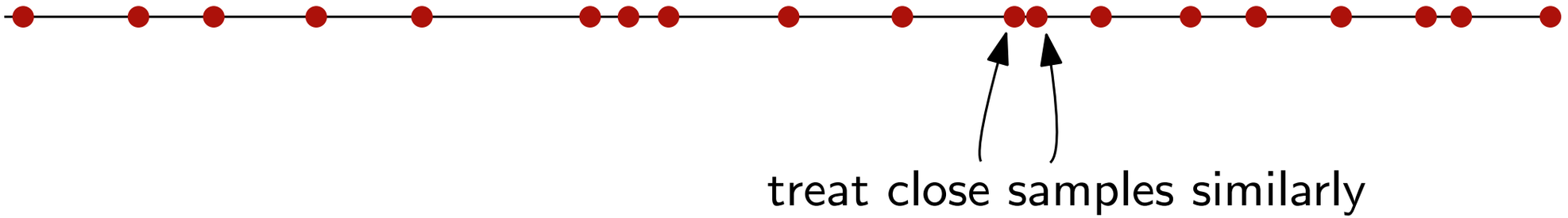
3) Randomization

Requirements of Sample Selection Algorithm



1) Closeness
2) Diversity
3) Randomization

Requirements of Sample Selection Algorithm



1) Closeness
2) Diversity
3) Randomization
4) Locality

Our Solution

Idea: Quantify all requirements and create function to be optimized

Our Solution

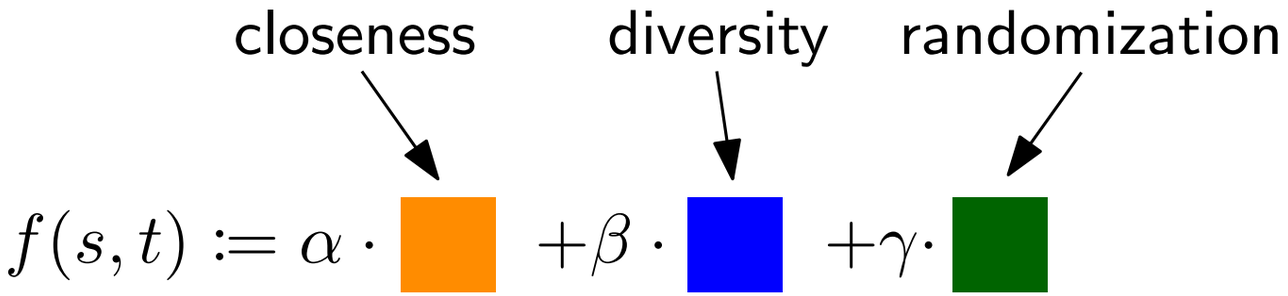
Idea: Quantify all requirements and create function to be optimized

$\rightarrow$ on request power p at time point t:
for each sample $s \in S$ compute score $f(s, t)$ ← "the lower the better"

Our Solution

Idea: Quantify all requirements and create function to be optimized

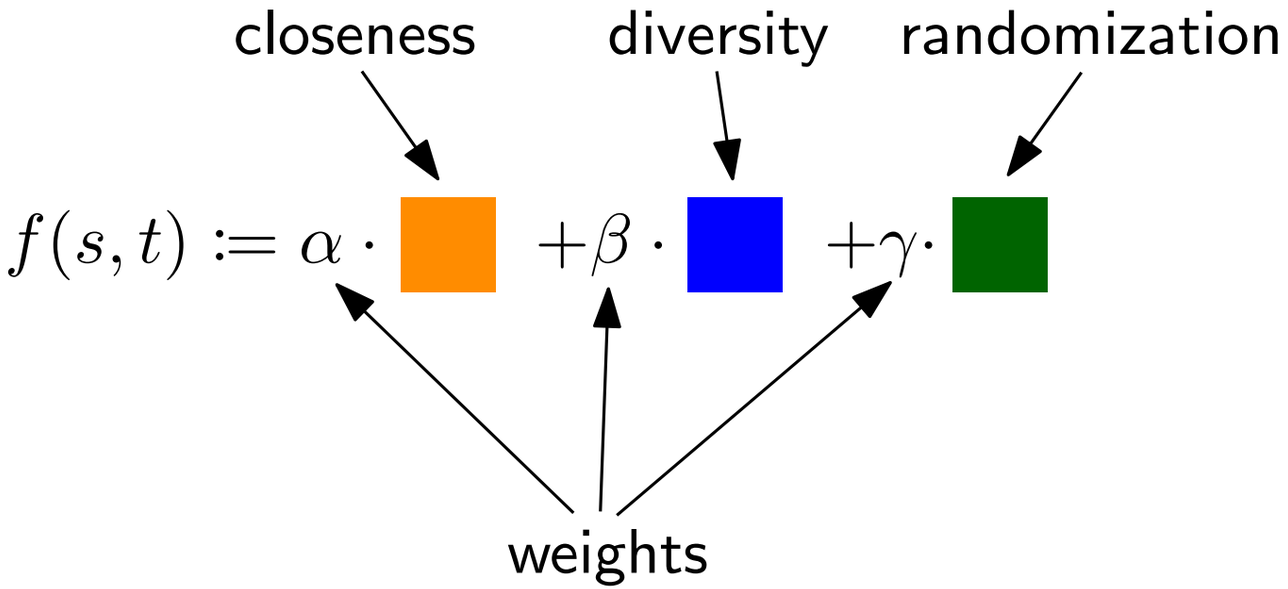
$\rightarrow$ on request power p at time point t:
for each sample $s \in S$ compute score $f(s,t)$ ← "the lower the better"



Our Solution

Idea: Quantify all requirements and create function to be optimized

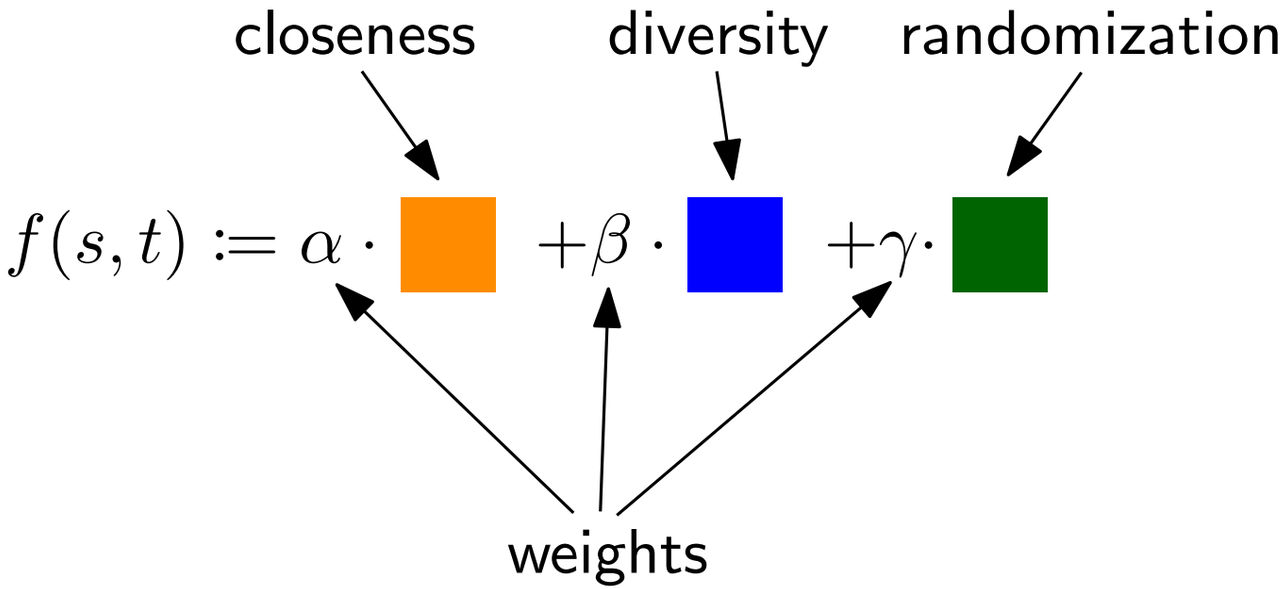
$\rightarrow$ on request power p at time point t:
for each sample $s \in S$ compute score $f(s,t)$ ← "the lower the better"



Our Solution

Idea: Quantify all requirements and create function to be optimized

$\rightarrow$ on request power p at time point t:
for each sample $s \in S$ compute score $f(s,t)$ ← "the lower the better"



$= \left(\frac{p - p_s}{p_{\max} - p_{\min}} \right)^2$

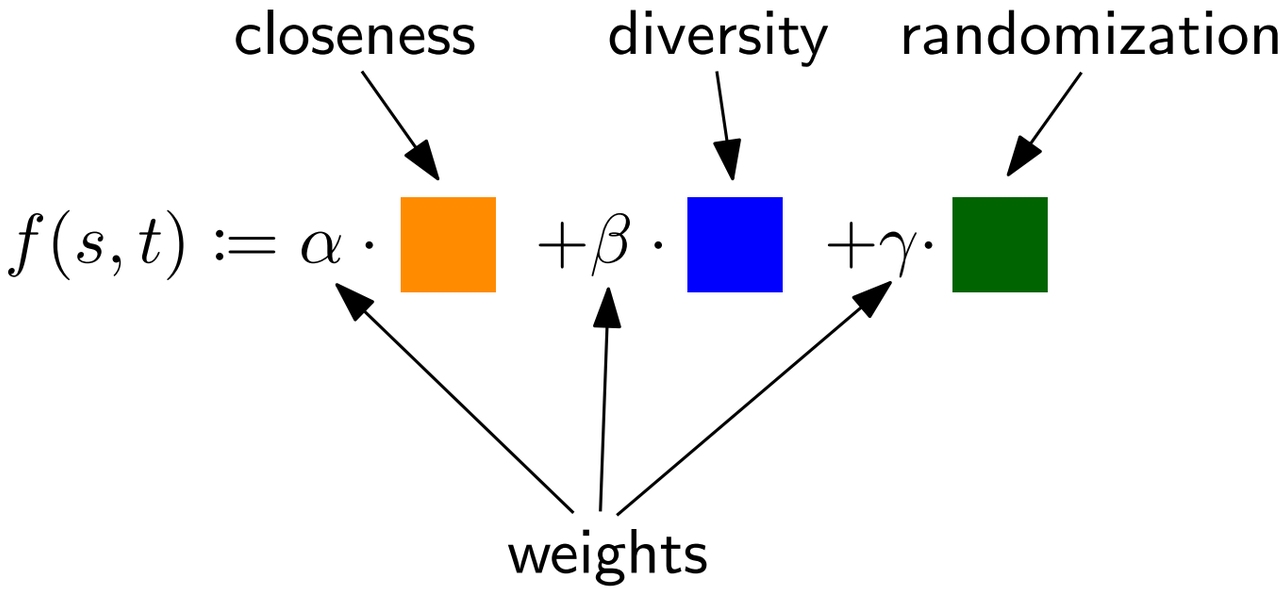
$= \left(1 + \frac{t - t_s}{S}\right)^{-1}$

$= r(s,t)$

Our Solution

Idea: Quantify all requirements and create function to be optimized

→ on request power p at time point t:
for each sample $s \in S$ compute score $f(s,t)$ ← "the lower the better"



$= \left(\frac{p - p_s}{p_{\max} - p_{\min}} \right)^2$

$= \left(1 + \frac{t - t_s}{S}\right)^{-1}$

$= r(s,t)$

→ **Are the requirements fulfilled?**

Our Solution

“Algorithm”

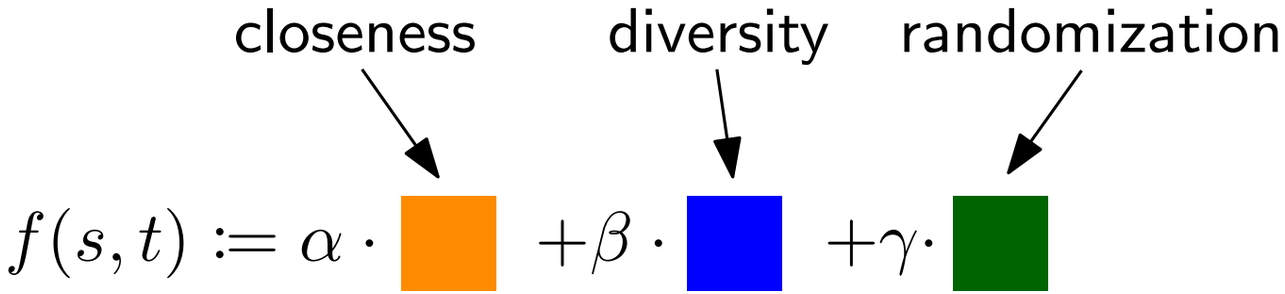
Input: Requested power p, instrument i, current time step t, parameters α, β, γ, and values t_s which are the time points sample s was played the last time

Output: Sample s

return $\arg\min_{s \in S_i} \alpha \cdot \left(\frac{p - p_s}{p_{\max} - p_{\min}} \right)^2 + \beta \cdot \left(1 + \frac{t - t_s}{S}\right)^{-1} + \gamma \cdot r(s, t)$

Emulation

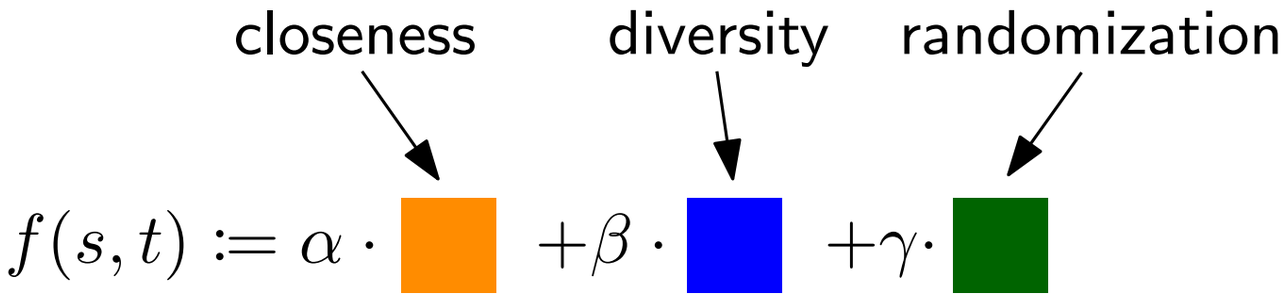
Three Extreme Cases



Choose closest sample: $\alpha = 1$ and $\beta = \gamma = 0$

Emulation

Three Extreme Cases

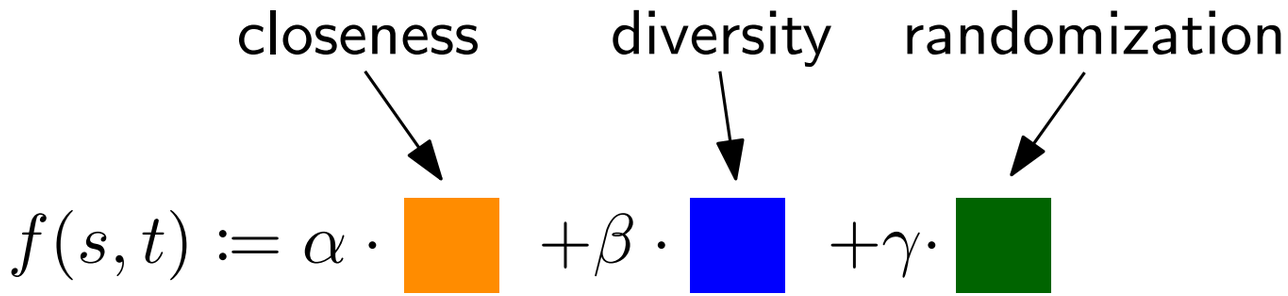


Choose closest sample: $\alpha = 1$ and $\beta = \gamma = 0$

Choose oldest: $\beta = 1$ and $\alpha = \gamma = 0$

Emulation

Three Extreme Cases



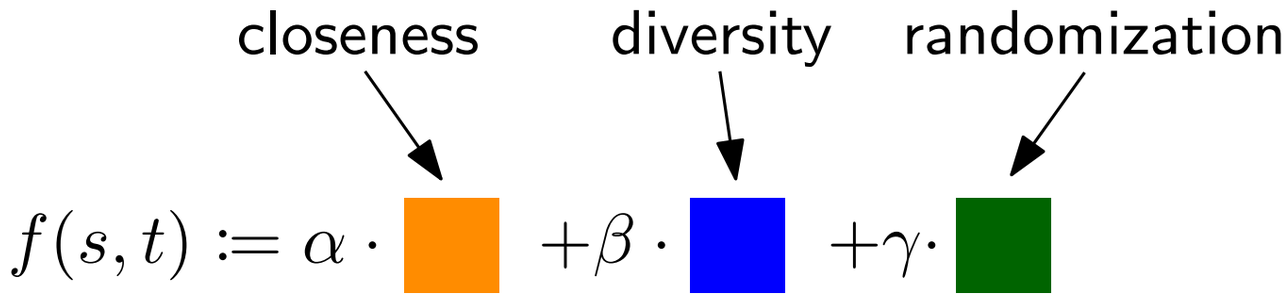
Choose closest sample: $\alpha = 1$ and $\beta = \gamma = 0$

Choose oldest: $\beta = 1$ and $\alpha = \gamma = 0$

Choose randomly: $\gamma = 1$ and $\alpha = \beta = 0$

Emulation

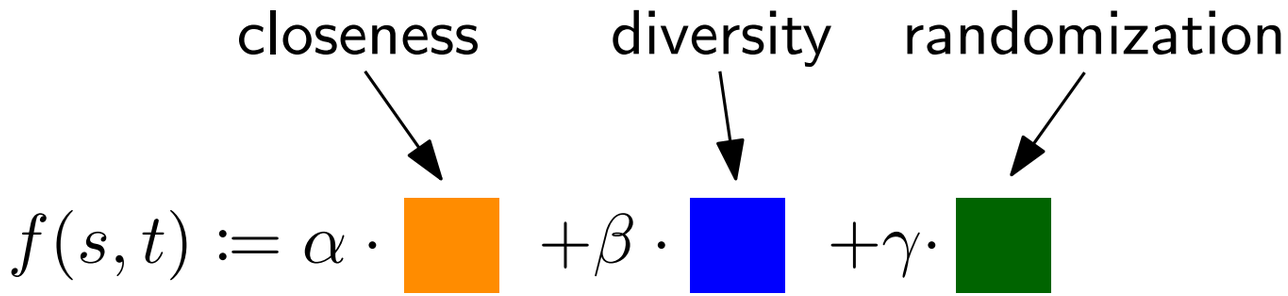
Round Robin



Prepare drum kit: Samples in same group have same power values

Emulation

Round Robin



Prepare drum kit: Samples in same group have same power values

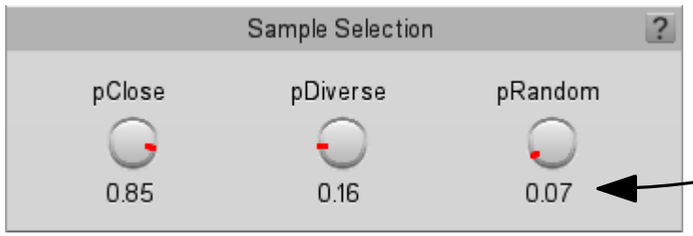
First choose closest: $\alpha =$ "large"

Then choose oldest: $\beta =$ "small"

And randomize a bit: $\gamma =$ "a bit smaller"

Implementation

- Introduced in DrumGizmo (in a preliminary form) in July 2019
- Simple controls:



values are not α, β, γ but some scaled version!

- Two improvement releases after user feedback

Experiments

Setting

Data:

- Crocell drum kit snare
- 98 samples (i.e., also power values)

Experiments:

- Repeatedly play same note
- Sweep MDI velocities several times

Experiments

Setting

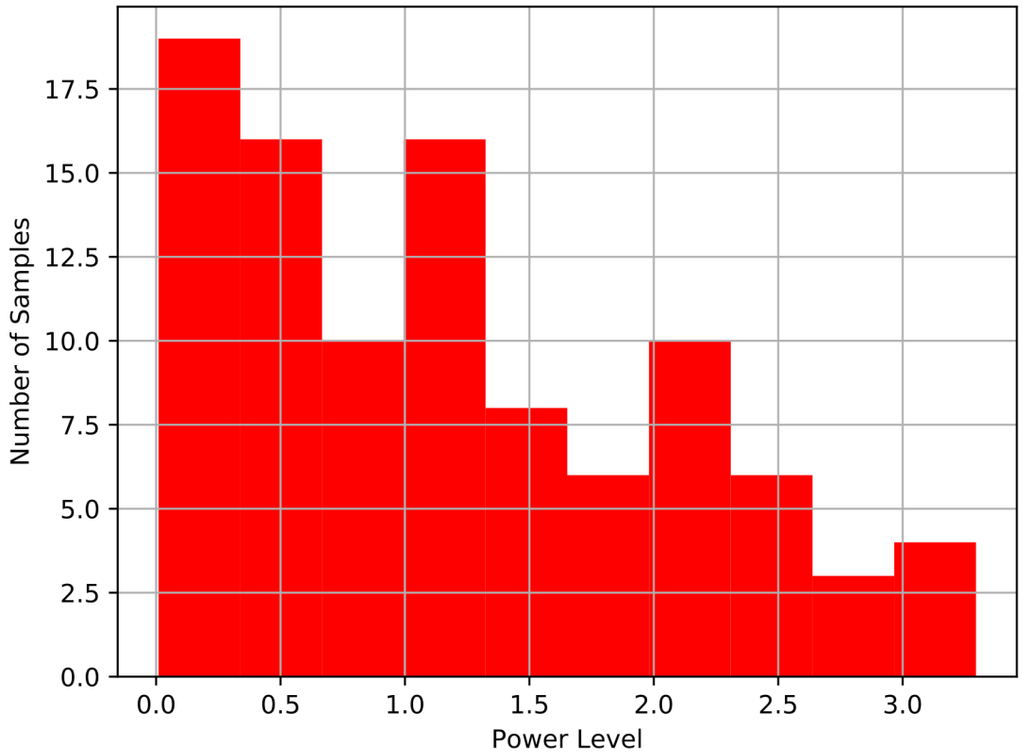
Data:

- Crocell drum kit snare
- 98 samples (i.e., also power values)

Experiments:

- Repeatedly play same note
- Sweep MDI velocities several times

Crocell Kit Power Level Distribution:



Experiments

Setting

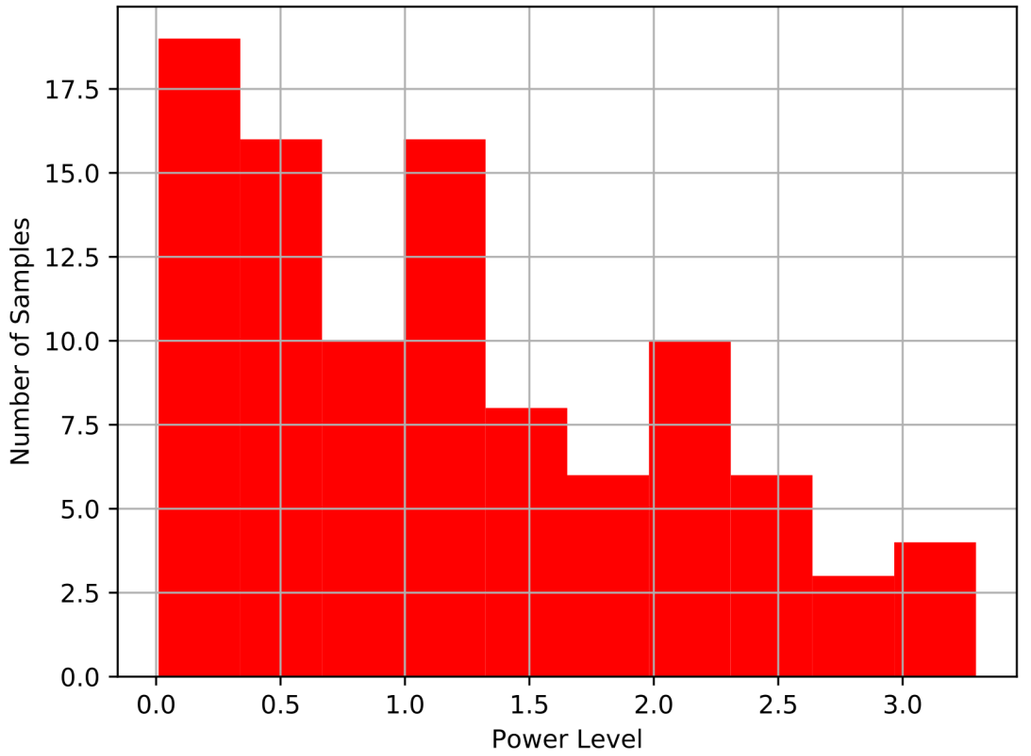
Data:

- Crocell drum kit snare
- 98 samples (i.e., also power values)

Experiments:

- Repeatedly play same note
- Sweep MDI velocities several times

Crocell Kit Power Level Distribution:

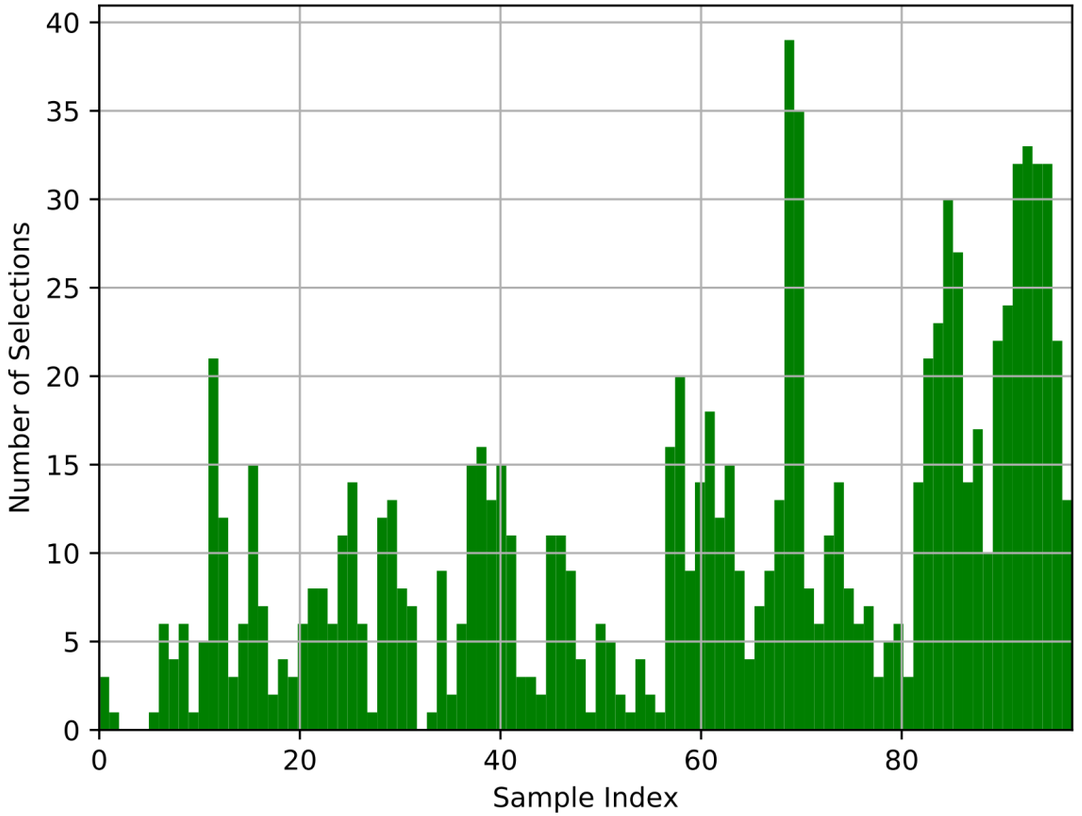


Why not sound tests?

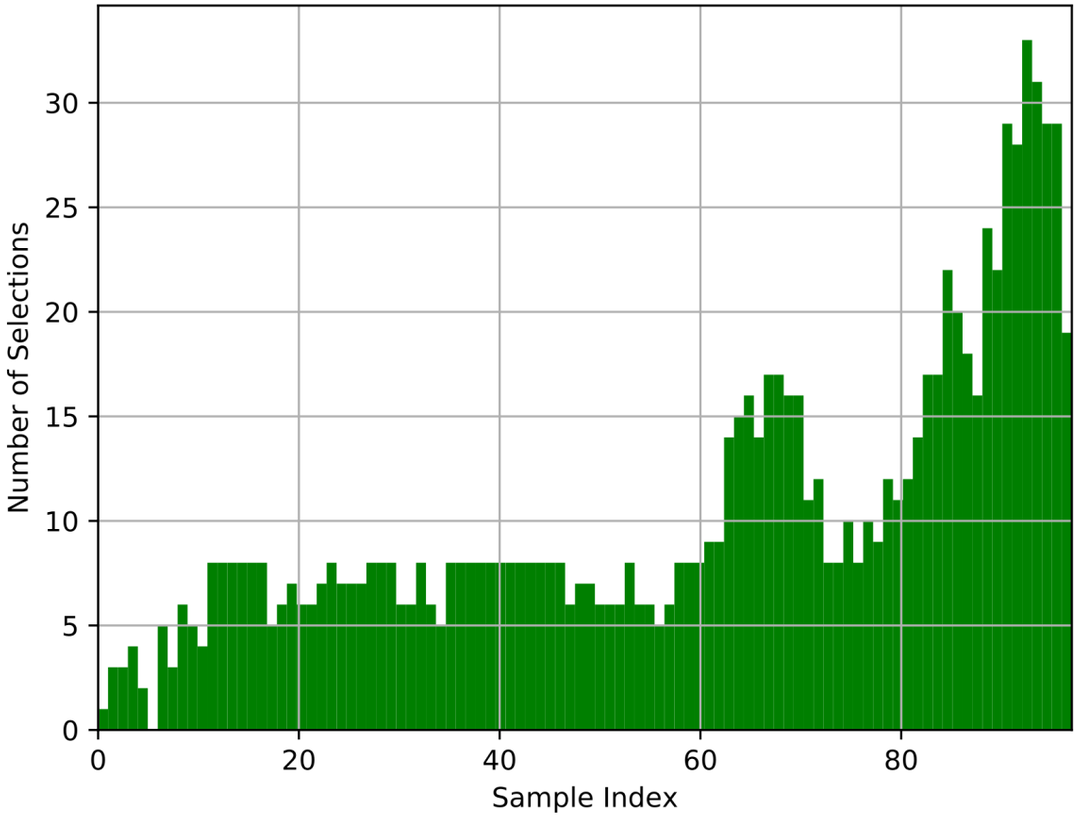
Experiments

Sweep

Old:



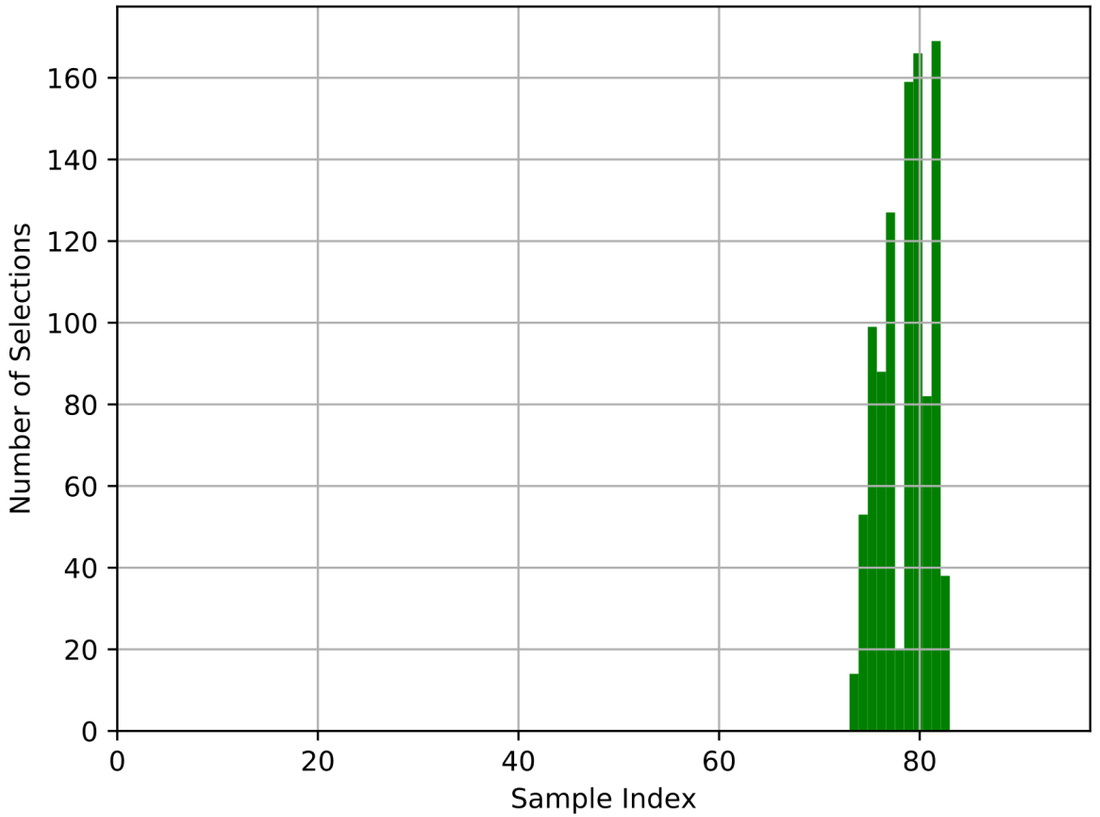
New:



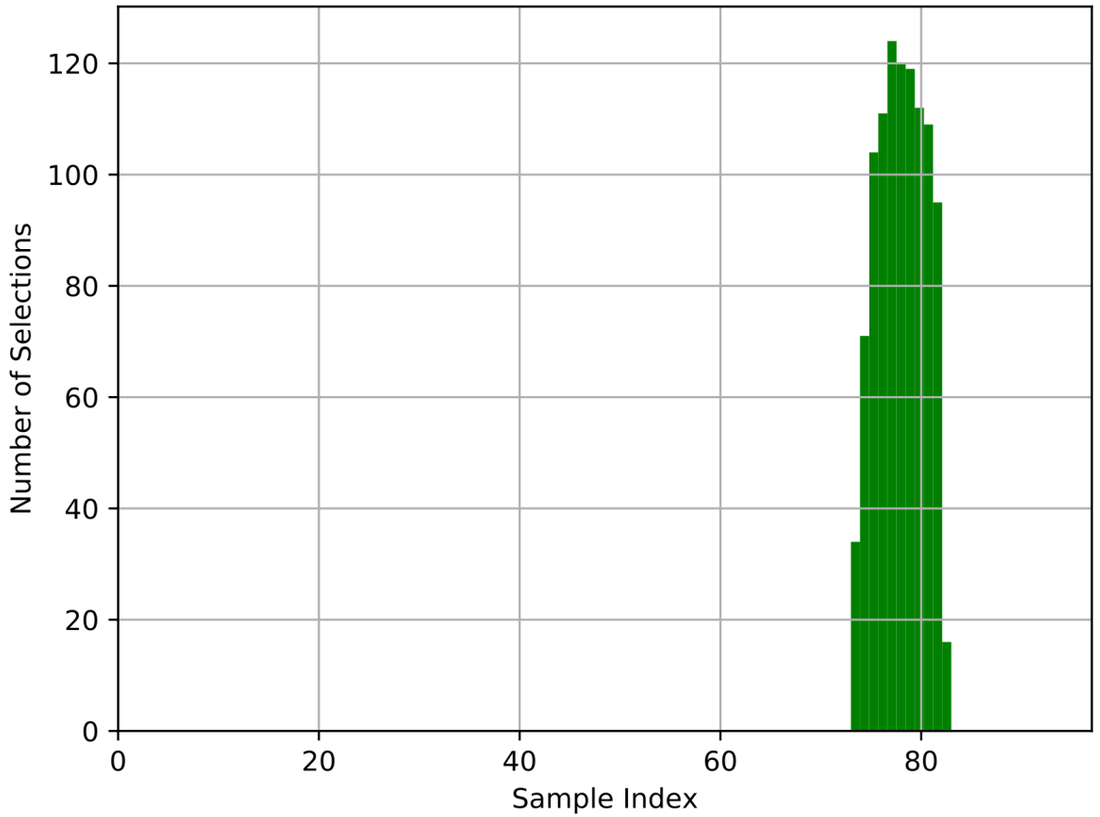
Experiments

Dense Sampling: MIDI Velocity 80

Old:



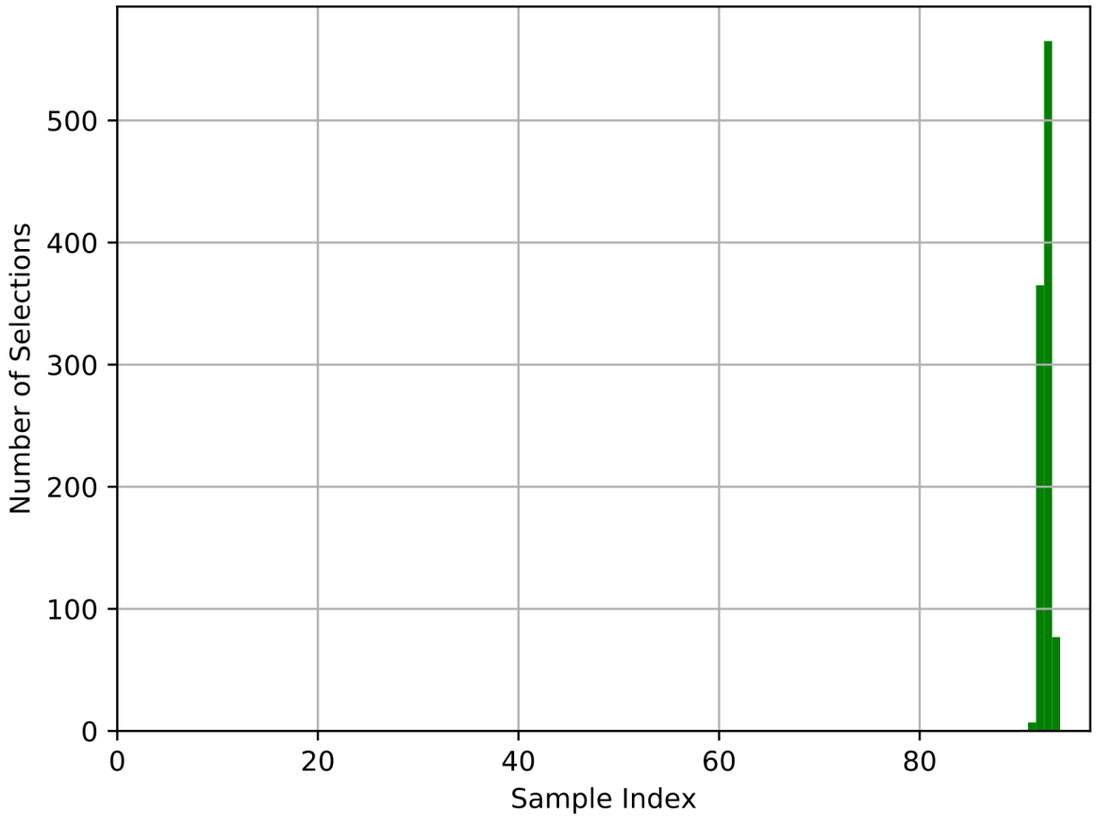
New:



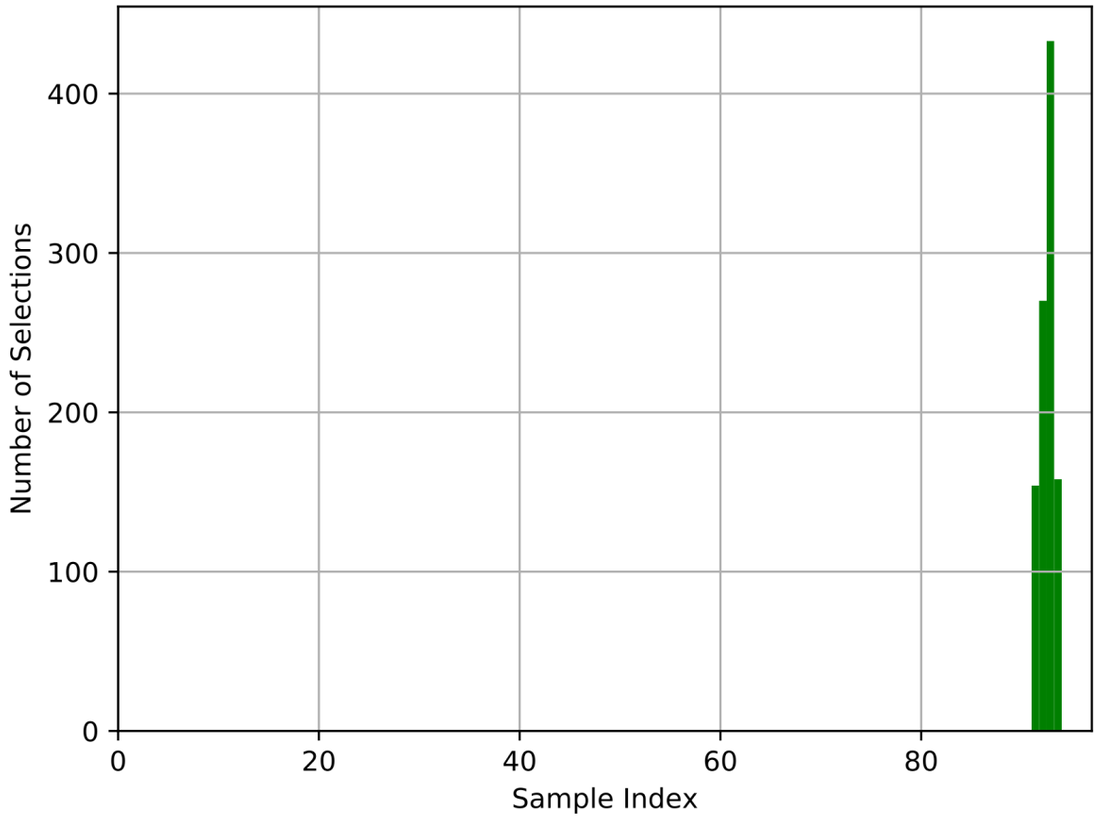
Experiments

Sparse Sampling: MIDI Velocity 112

Old:



New:



Experiments

User Feedback

- First Example:
  - **Issue:** Fast notes on sparsely sampled cymbal
  - **Solution:** Change diversity term from quadratic to linear

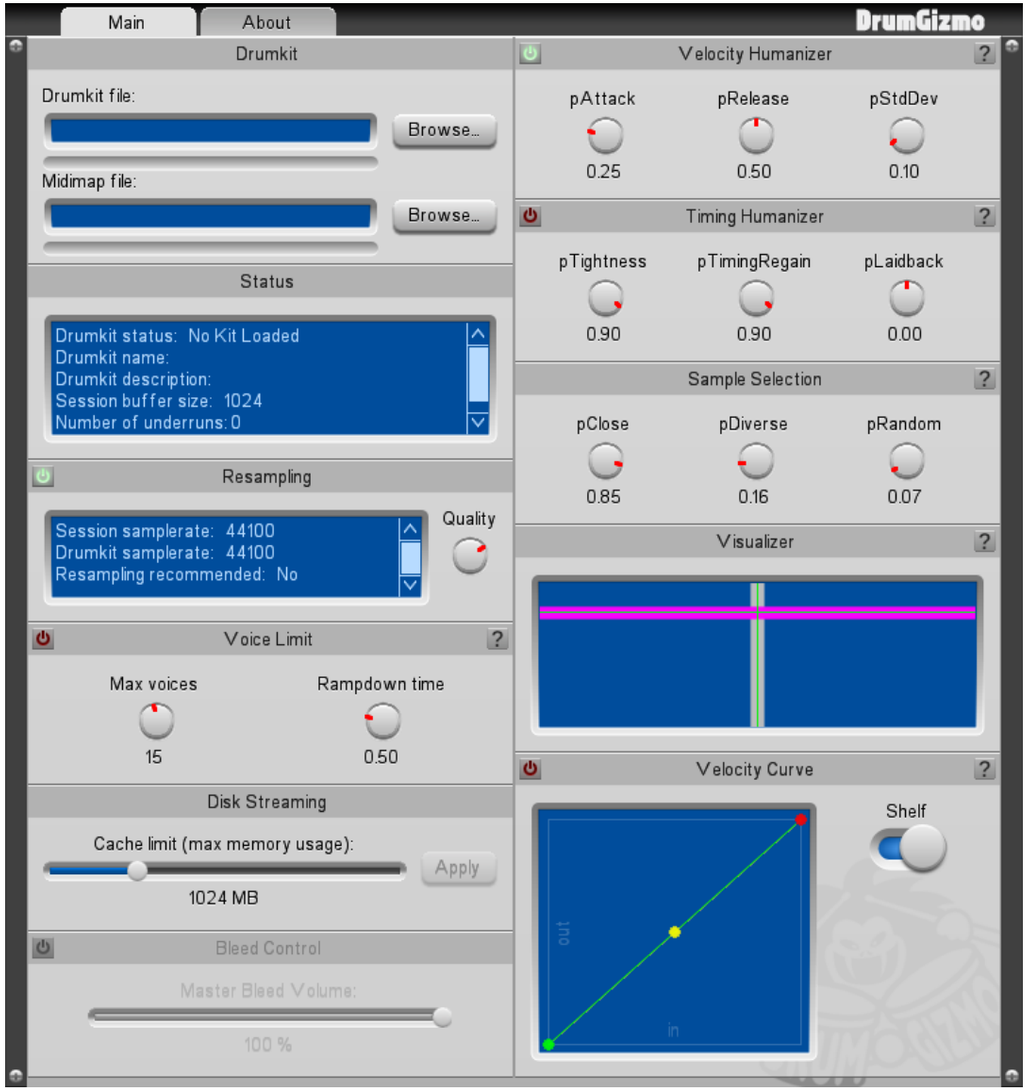
Experiments

User Feedback

- First Example:
  - **Issue:** Fast notes on sparsely sampled cymbal
  - **Solution:** Change diversity term from quadratic to linear
- Second Example:
  - **Issue:** Suboptimal performance on certain drum kit
  - **Quick Solution:** Change default parameters
  - **Long-Term Solution:** Each drum kit provides its parameters

Final Remarks

New Release!



Thanks for your attention!

www.drumgizmo.org