# CMA-ES with Learning Rate Adaptation: Can CMA-ES with Default Population Size Solve Multimodal and Noisy Problems?

GECCO'23

**Masahiro Nomura** (Tokyo Institute of Technology)
Youhei Akimoto (University of Tsukuba & RIKEN AIP)
Isao Ono (Tokyo Institute of Technology)

# Outline

1. CMA-ES and Its Issues for **difficult (multimodal and/or noisy) problems**

   Users need **expensive** hyperparameter tuning (e.g. population size)

# Outline

1. CMA-ES and Its Issues for **difficult (multimodal and/or noisy) problems**

   Users need **expensive** hyperparameter tuning (e.g. population size)

2. CMA-ES with **Learning Rate Adaptation**

   Can the CMA-ES with **default** population size (λ) solve multimodal and noisy problems?

# Outline

1. CMA-ES and Its Issues for **difficult (multimodal and/or noisy) problems**

   Users need **expensive** hyperparameter tuning (e.g. population size)

2. CMA-ES with **Learning Rate Adaptation**

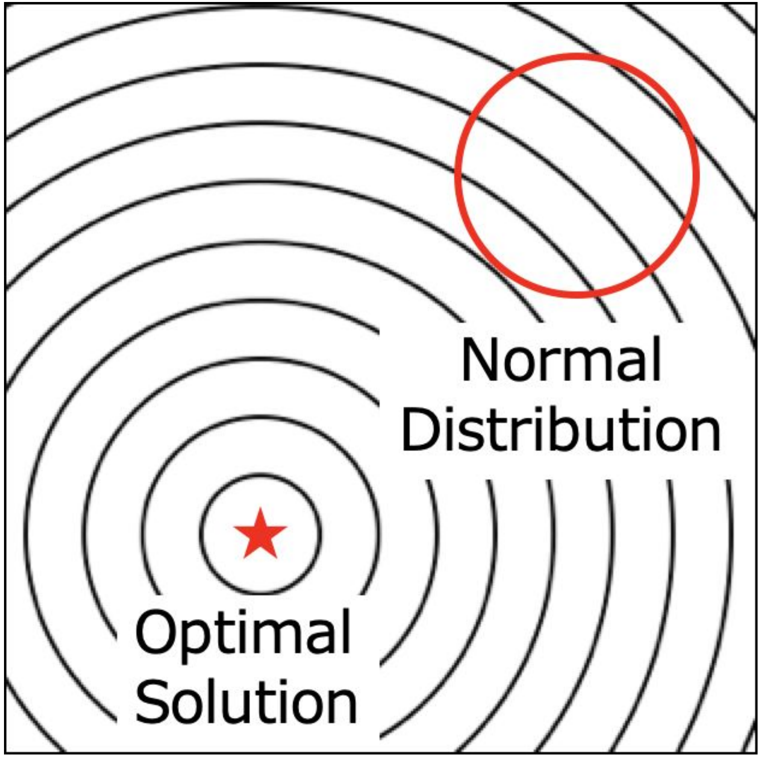
   Can the CMA-ES with **default** population size ($\lambda$) solve multimodal and noisy problems?

3. Experimental Results

   With LRA, CMA with **default** $\lambda$ (e.g. $\lambda$=15 for d=40) can succeed on Rastrigin

# CMA-ES [HMK03,Han16]

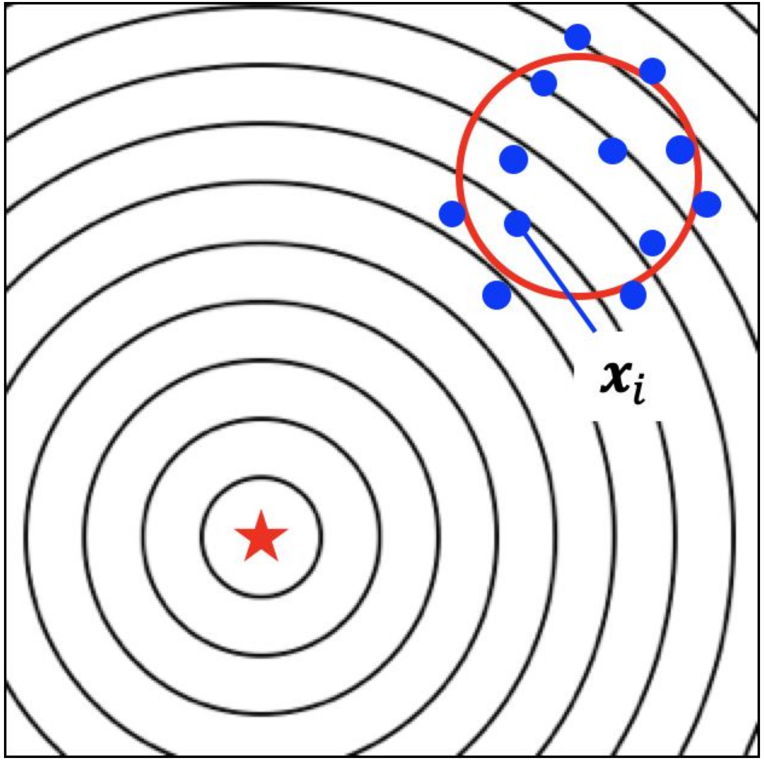
- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$



# CMA-ES [HMK03,Han16]

- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$

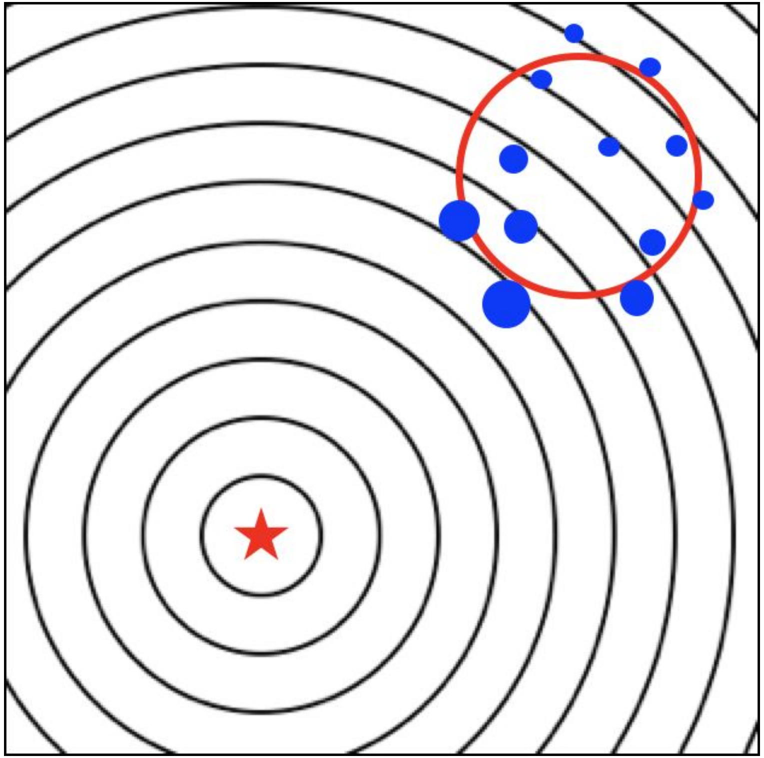
1. Generates solutions from the MGD



# CMA-ES [HMK03,Han16]

- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$

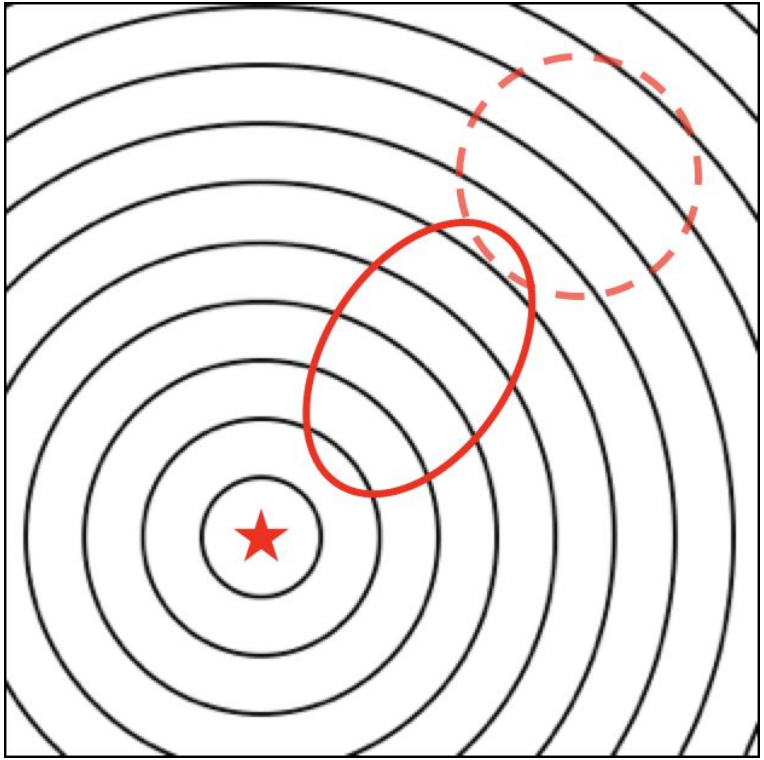
1. Generates solutions from the MGD
2. Evaluates and weights solutions



# CMA-ES [HMK03,Han16]

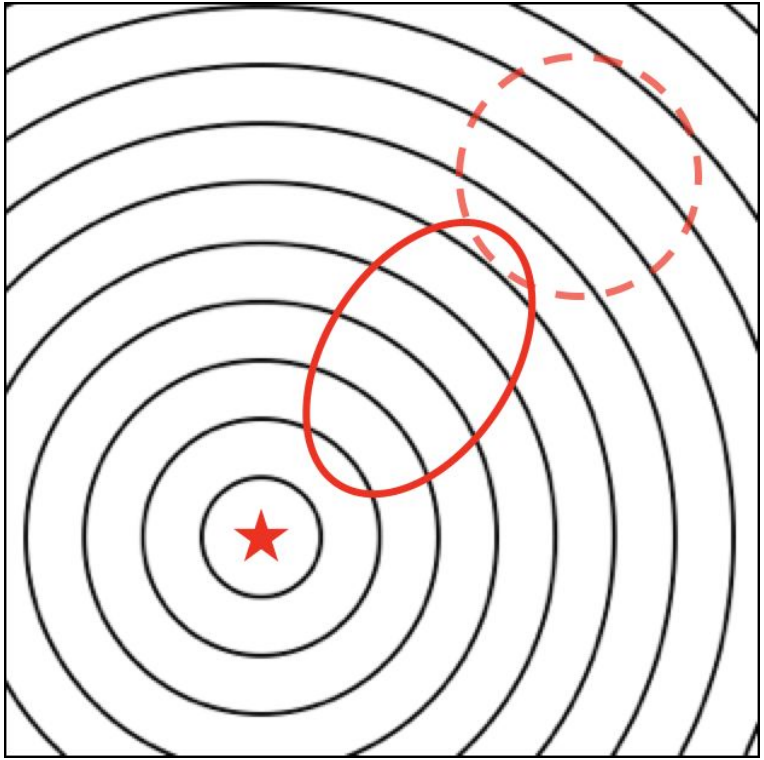
- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$

1. Generates solutions from the MGD
2. Evaluates and weights solutions
3. Updates the parameters



# CMA-ES [HMK03,Han16]

- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$

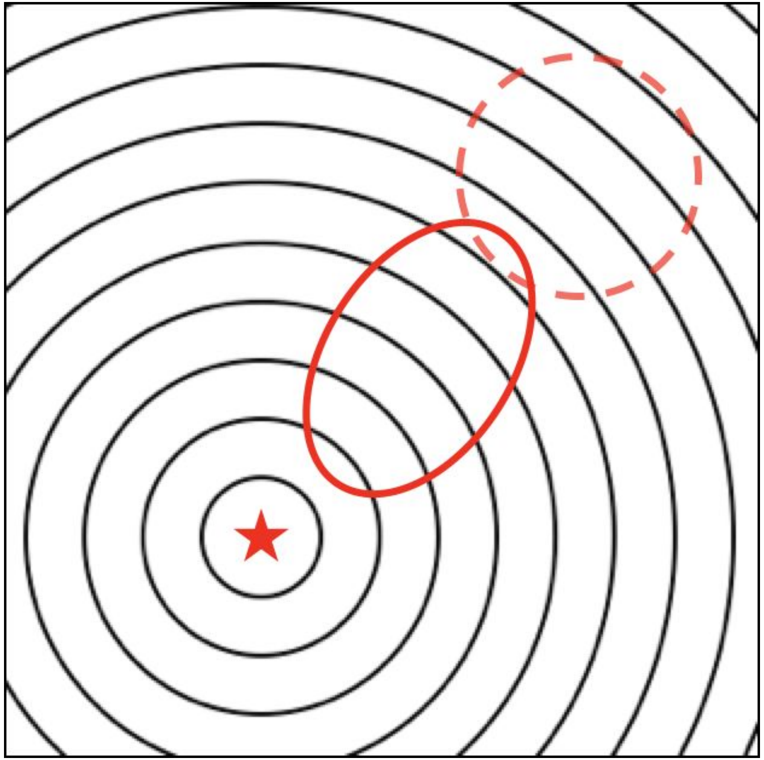


1. Generates solutions from the MGD
2. Evaluates and weights solutions
3. Updates the parameters
4. Repeats until the criterion is met

# CMA-ES [HMK03,Han16]

- one of the most promising BBO methods
- multivariate Gaussian distribution (MGD)
  - parameterized by $\mathcal{N}(m, \sigma^2 C)$

1. Generates solutions from the MGD
2. Evaluates and weights solutions
3. Updates the parameters
4. Repeats until the criterion is met



We consider the [most commonly used CMA-ES]

# CMA-ES and Its Dependence on Hyperparameters

- CMA-ES is a *quasi-hyperparameter-free* method
  - Hyperparameter values are automatically computed from:
    - (1) dimensionality, (2) ***population size λ***; by default, $\lambda = 4 + \lfloor 3 \ln d \rfloor$
  - Default λ works well for relatively ***easy*** problems

# CMA-ES and Its Dependence on Hyperparameters

- CMA-ES is a <u>*quasi-hyperparameter-free*</u> method
  - Hyperparameter values are automatically computed from:
    - (1) dimensionality, (2) <u>***population size λ***</u>; by default, $\lambda = 4 + \lfloor 3 \ln d \rfloor$
  - Default λ works well for relatively <u>***easy***</u> problems
- Increasing (i.e. not default) λ can be helpful for <u>***difficult***</u> problems
  - Knowing good value is challenging ⇒ <u>expensive tuning is required</u>

# CMA-ES and Its Dependence on Hyperparameters

- CMA-ES is a *quasi-hyperparameter-free* method
  - Hyperparameter values are automatically computed from:
    - (1) dimensionality, (2) ***population size λ***; by default, $\lambda = 4 + \lfloor 3 \ln d \rfloor$
  - Default λ works well for relatively ***easy*** problems
- Increasing (i.e. not default) λ can be helpful for ***difficult*** problems
  - Knowing good value is challenging ⇒ expensive tuning is required
- Possible approach: online λ adaptation [NA16,NA18]

# Our Approach: Learning Rate Adaptation

- Important observation [MA17]:
  - Increasing $\lambda$ has effect similar to decreasing mean-vector learning rate
  - CMA-ES with a small population size solves multimodal problems
    **if the learning rate setting is appropriate**

# Our Approach: Learning Rate Adaptation

- Important observation [MA17]:
  - Increasing λ has effect similar to decreasing mean-vector learning rate
  - CMA-ES with a small population size solves multimodal problems
    if the learning rate setting is appropriate
- **Learning rate adaptation** vs. **population size adaptation**
  - Learning rate adaptation is more practically useful
  - E.g. parallel implementation (may be *population size = # of workers*)

# Learning Rate Adaptation: Setup

vectorization operator, Σ = σ^2 C

- Notations :
  - distribution parameters : $\theta_m = m, \theta_\Sigma = \underline{\mathrm{vec}(\Sigma)}$
  - original updates : $\Delta_m^{(t)} = m^{(t+1)} - m^{(t)}, \Delta_\Sigma^{(t)} = \mathrm{vec}(\Sigma^{(t+1)} - \Sigma^{(t)})$
  - learning rate factors : $\eta_m^{(t)}, \eta_\Delta^{(t)}$

# Learning Rate Adaptation: Setup

vectorization operator, Σ = σ^2 C

- Notations :
  - distribution parameters : $\theta_m = m, \theta_\Sigma = \underline{\mathrm{vec}(\Sigma)}$
  - original updates : $\Delta_m^{(t)} = m^{(t+1)} - m^{(t)}, \Delta_\Sigma^{(t)} = \mathrm{vec}(\Sigma^{(t+1)} - \Sigma^{(t)})$
  - learning rate factors : $\eta_m^{(t)}, \eta_\Delta^{(t)}$
- Modified updates :
  - $\theta_m^{(t+1)} = \theta_m^{(t)} + \eta_m^{(t)} \Delta_m^{(t)}$
  - $\theta_\Sigma^{(t+1)} = \theta_\Sigma^{(t)} + \eta_\Sigma^{(t)} \Delta_\Sigma^{(t)}$

original updates can be recovered with $\eta$ =1

# Learning Rate Adaptation: Setup

vectorization operator, Σ = σ^2 C

- Notations :
  - distribution parameters : $\theta_m = m, \theta_\Sigma = \underline{\mathrm{vec}(\Sigma)}$
  - original updates : $\Delta_m^{(t)} = m^{(t+1)} - m^{(t)}, \Delta_\Sigma^{(t)} = \mathrm{vec}(\Sigma^{(t+1)} - \Sigma^{(t)})$
  - learning rate factors : $\eta_m^{(t)}, \eta_\Delta^{(t)}$
- Modified updates :
  - $\theta_m^{(t+1)} = \theta_m^{(t)} + \boxed{\eta_m^{(t)}} \Delta_m^{(t)}$
  - $\theta_\Sigma^{(t+1)} = \theta_\Sigma^{(t)} + \boxed{\eta_\Sigma^{(t)}} \Delta_\Sigma^{(t)}$

original updates can be recovered with η =1

*How to adapt these learning rate?*

# Why We Use SNR for Learning Rate Adaptation?

- We adapt the learning rate based on the ***signal-to-noise ratio (SNR)***:

$$\mathrm{SNR} := \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathrm{Tr}(F\,\mathrm{Cov}[\Delta])} = \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathbb{E}[\|\Delta\|_F^2] - \|\mathbb{E}[\Delta]\|_F^2}$$

*F*: Fisher information matrix

# Why We Use SNR for Learning Rate Adaptation?

- We adapt the learning rate based on the ***signal-to-noise ratio (SNR)***:

$$\mathrm{SNR} := \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathrm{Tr}(F\,\mathrm{Cov}[\Delta])} = \frac{\|\mathbb{E}[\Delta]\|_F^2}{\mathbb{E}[\|\Delta\|_F^2] - \|\mathbb{E}[\Delta]\|_F^2}$$

*F*: Fisher information matrix

- ***Noisy problems***: ***SNR → 0 when noise becomes dominant***
  - To improve function value, ***maintaining a positive SNR is crucial***
  - We apply similar arguments to ***multimodal problems***

# SNR-Based Learning Rate Adaptation

- Assume LR is small over *n* iterations ⇔ updates are i.i.d
- *n* steps update:

$$
\begin{aligned}
\theta^{(t+n)} &= \theta^{(t)} + \eta \sum_{k=0}^{n-1} \Delta^{(t+k)} \\
&\approx \theta^{(t)} + \mathcal{D}\left(n\eta\mathbb{E}[\Delta], n\eta^2 \mathrm{Cov}[\Delta]\right)
\end{aligned}
$$

# SNR-Based Learning Rate Adaptation

- Assume LR is small over *n* iterations ⇔ updates are i.i.d
- *n* steps update:

$$\theta^{(t+n)} = \theta^{(t)} + \eta \sum_{k=0}^{n-1} \Delta^{(t+k)}$$

$$\approx \theta^{(t)} + \boxed{\mathcal{D}\left(n\eta\mathbb{E}[\Delta], n\eta^2 \mathrm{Cov}[\Delta]\right)}$$

- $n = 1/\eta \Rightarrow \mathcal{D}\left(\mathbb{E}[\Delta], \eta\mathrm{Cov}[\Delta]\right)$
  - By taking small $\eta$, we can obtain *more concentrated update*

# SNR-Based Learning Rate Adaptation

- Assume LR is small over *n* iterations ⇔ updates are i.i.d
- *n* steps update:

$$\theta^{(t+n)} = \theta^{(t)} + \eta \sum_{k=0}^{n-1} \Delta^{(t+k)}$$
$$\approx \theta^{(t)} + \mathcal{D}\left(n\eta\mathbb{E}[\Delta], n\eta^2\mathrm{Cov}[\Delta]\right)$$

- *n = 1/η* ⇒ $\mathcal{D}\left(\mathbb{E}[\Delta], \eta\mathrm{Cov}[\Delta]\right)$
  - By taking small *η*, we can obtain *more concentrated update*
- SNR over *n* iterations: $\dfrac{\|\mathbb{E}[\Delta]\|_F^2}{\eta\,\mathrm{Tr}(F\,\mathrm{Cov}[\Delta])} = \dfrac{1}{\eta}\mathrm{SNR}$
- **Our method**: keep *SNR over n(=1/η) itr.* as *(positive) constant*
  - $\mathrm{SNR} = \alpha\eta \quad (\alpha > 0)$

# SNR Estimation with Moving Averages

- We introduce moving averages for each m and Σ

$$\mathcal{E}^{(t+1)} = (1-\beta)\mathcal{E}^{(t)} + \beta\tilde{\Delta}^{(t)},$$

$$\mathcal{V}^{(t+1)} = (1-\beta)\mathcal{V}^{(t)} + \beta\|\tilde{\Delta}^{(t)}\|_2^2$$

local coordinate

$$\tilde{\Delta}_m = \sqrt{\Sigma}^{-1}\Delta_m$$

$$\tilde{\Delta}_\Sigma = 2^{-\frac{1}{2}}\mathrm{vec}(\sqrt{\Sigma}^{-1}\mathrm{vec}^{-1}(\Delta_\Sigma)\sqrt{\Sigma}^{-1})$$

# SNR Estimation with Moving Averages

- We introduce moving averages for each m and Σ

$$\mathcal{E}^{(t+1)} = (1-\beta)\mathcal{E}^{(t)} + \beta\tilde{\Delta}^{(t)},$$

$$\mathcal{V}^{(t+1)} = (1-\beta)\mathcal{V}^{(t)} + \beta\|\tilde{\Delta}^{(t)}\|_2^2$$

local coordinate

$$\tilde{\Delta}_m = \sqrt{\Sigma}^{-1}\Delta_m$$

$$\tilde{\Delta}_\Sigma = 2^{-\frac{1}{2}}\mathrm{vec}(\sqrt{\Sigma}^{-1}\mathrm{vec}^{-1}(\Delta_\Sigma)\sqrt{\Sigma}^{-1})$$

- The SNR is estimated as:

$$\mathrm{SNR} := \frac{\mathbb{E}[\tilde{\Delta}]^2}{\mathrm{Tr}(\mathrm{Cov}[\tilde{\Delta}])} = \frac{\mathbb{E}[\tilde{\Delta}]^2}{\mathbb{E}[\|\tilde{\Delta}\|^2] - \|\mathbb{E}[\tilde{\Delta}]\|^2},$$

$$\approx \frac{\|\mathcal{E}\|_2^2 - \frac{\beta}{2-\beta}\mathcal{V}}{\mathcal{V} - \|\mathcal{E}\|_2^2} =: \widehat{\mathrm{SNR}}$$

# SNR Estimation with Moving Averages

- We introduce moving averages for each m and Σ

$$\mathcal{E}^{(t+1)} = (1-\beta)\mathcal{E}^{(t)} + \beta\tilde{\Delta}^{(t)},$$
$$\mathcal{V}^{(t+1)} = (1-\beta)\mathcal{V}^{(t)} + \beta\|\tilde{\Delta}^{(t)}\|_2^2$$

local coordinate

$$\tilde{\Delta}_m = \sqrt{\Sigma}^{-1}\Delta_m$$
$$\tilde{\Delta}_\Sigma = 2^{-\frac{1}{2}}\mathrm{vec}(\sqrt{\Sigma}^{-1}\mathrm{vec}^{-1}(\Delta_\Sigma)\sqrt{\Sigma}^{-1})$$

- The SNR is estimated as:

$$\mathrm{SNR} := \frac{\mathbb{E}[\tilde{\Delta}]^2}{\mathrm{Tr}(\mathrm{Cov}[\tilde{\Delta}])} = \frac{\mathbb{E}[\tilde{\Delta}]^2}{\mathbb{E}[\|\tilde{\Delta}\|^2] - \|\mathbb{E}[\tilde{\Delta}]\|^2},$$

$$\approx \frac{\|\mathcal{E}\|_2^2 - \frac{\beta}{2-\beta}\mathcal{V}}{\mathcal{V} - \|\mathcal{E}\|_2^2} =: \widehat{\mathrm{SNR}}$$

Approximation!
(See paper for details)

# Update Equation of Learning Rate Adaptation

Adapting learning rate by:

$$\eta \leftarrow \eta \cdot \exp\left(\min(\gamma\eta, \beta)\Pi_{[-1,1]}\left(\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1\right)\right)$$

# Update Equation of Learning Rate Adaptation

Adapting learning rate by:

bring SNR closer to $\alpha\eta$

$$\eta \leftarrow \eta \cdot \exp\left(\min(\gamma\eta, \beta)\Pi_{[-1,1]}\left(\boxed{\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1}\right)\right)$$

# Update Equation of Learning Rate Adaptation

Adapting learning rate by:

bring SNR closer to $\alpha\eta$

$$\eta \leftarrow \eta \cdot \exp\left(\min(\gamma\eta, \beta)\Pi_{[-1,1]}\left(\boxed{\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1}\right)\right)$$

wait for the effect of the change of previous $\eta$

prevent $\eta$ to change more than the factor of exp($y$) or exp(-$y$) in 1/$\eta$ iterations

# Update Equation of Learning Rate Adaptation

Adapting learning rate by:

projection onto [-1, 1]

bring SNR closer to $\alpha\eta$

$$\eta \leftarrow \eta \cdot \exp\left(\min(\gamma\eta, \beta)\Pi_{[-1,1]}\left(\boxed{\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1}\right)\right)$$

wait for the effect of the change of previous $\eta$

prevent $\eta$ to change more than the factor of exp($\gamma$) or exp(-$\gamma$) in 1/$\eta$ iterations

# Update Equation of Learning Rate Adaptation

Adapting learning rate by:

projection onto [-1, 1]

bring SNR closer to $\alpha\eta$

$$\eta \leftarrow \eta \cdot \exp\left(\underbrace{\min(\gamma\eta,}_{}\ \underbrace{\beta)}_{}\ \underbrace{\Pi_{[-1,1]}}_{}\left(\boxed{\frac{\widehat{\mathrm{SNR}}}{\alpha\eta} - 1}\right)\right)$$

wait for the effect of the change of previous $\eta$

prevent $\eta$ to change more than the factor of exp($y$) or exp(-$y$) in $1/\eta$ iterations

$$\eta \leftarrow \min(\eta, \underbrace{1}_{})$$

upper bound

# Experiments: Research Questions and Setups

- **RQ1.** Does the learning rate adaptation (LRA) behave appropriately depending on search situations?
- **RQ2.** Can LRA-CMA (CMA-ES with LRA) with *default $\lambda$* solve multimodal and/or noisy problems?
- (See paper for **RQ3.** Hyperparameter Sensitivity)

# Experiments: Research Questions and Setups

- RQ1. Does the learning rate adaptation (LRA) behave appropriately depending on search situations?
- RQ2. Can LRA-CMA (CMA-ES with LRA) with *default $\lambda$* solve multimodal and/or noisy problems?
- (See paper for RQ3. Hyperparameter Sensitivity)

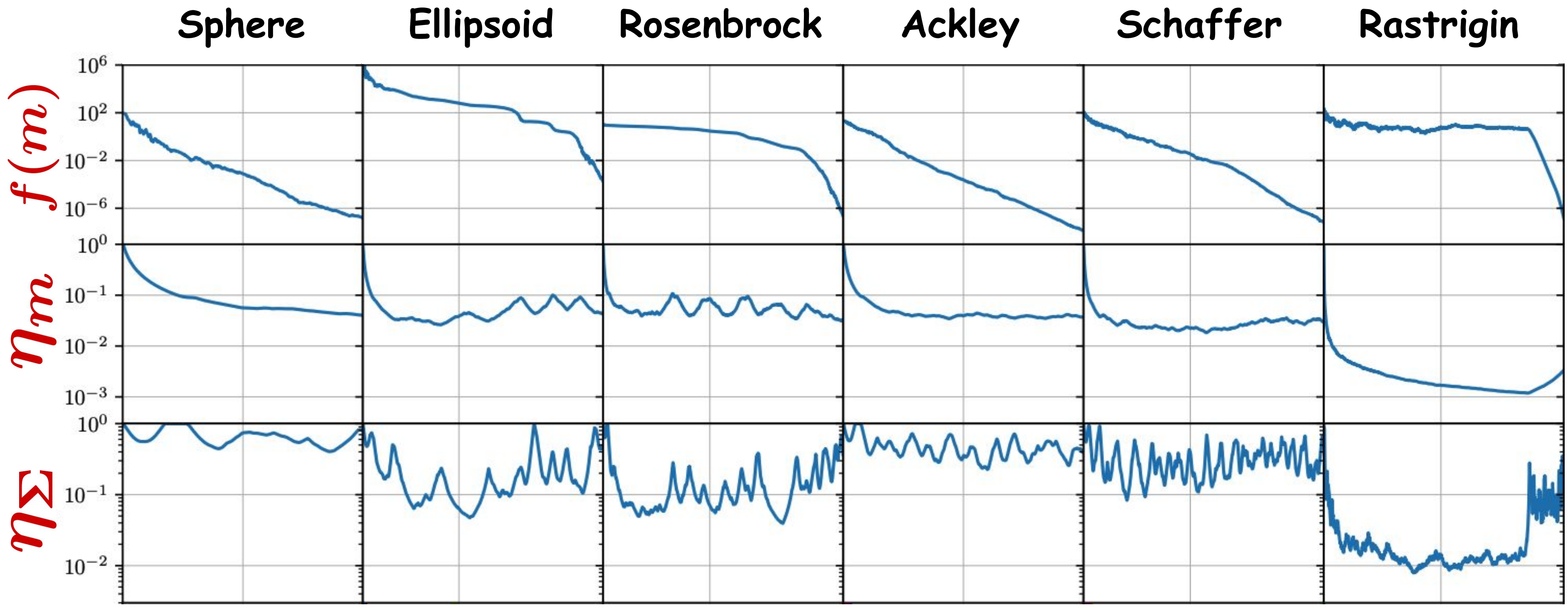
- Benchmark problems
  - 3 *unimodal* functions & 5 *multimodal* functions

Sphere, Ellipsoid, Rosenbrock

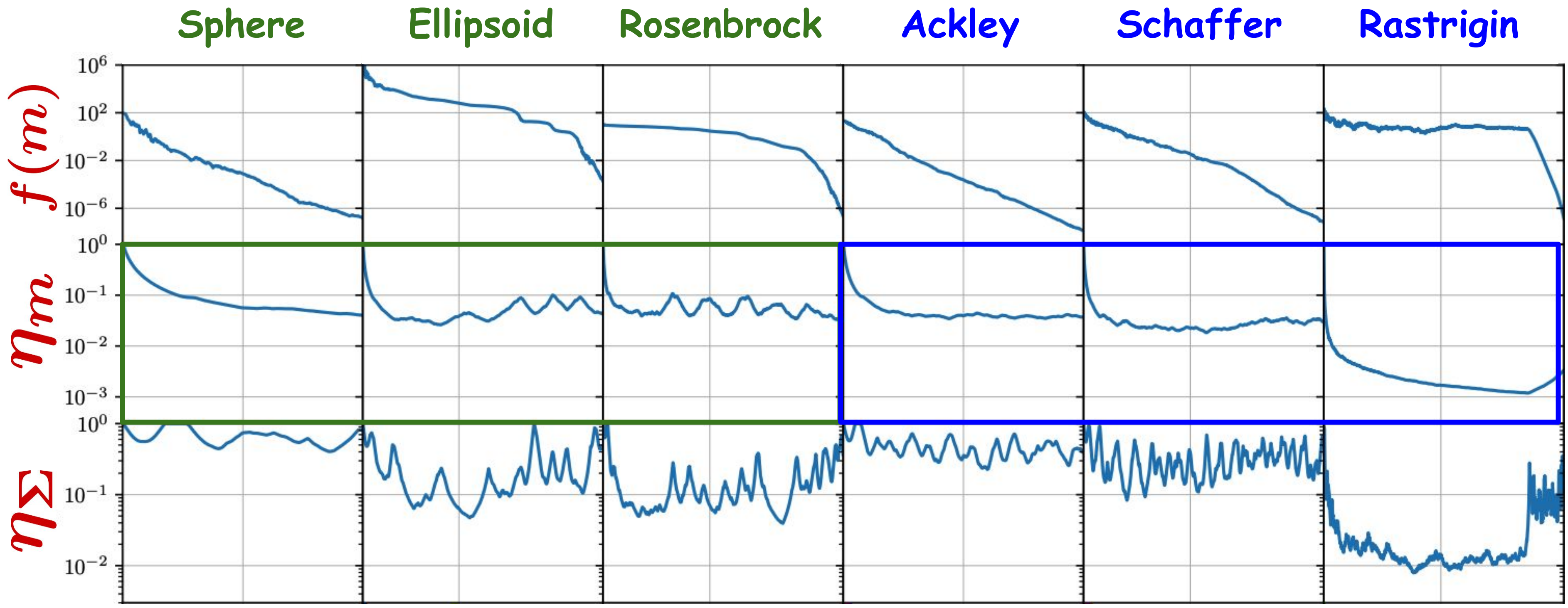
Ackley, Schaffer, Rastrigin, Bohachevsky, Griewank

  - In noisy problems, we considered additive Gaussian noise

# Typical LRA-CMA behavior on 10-D noiseless problems

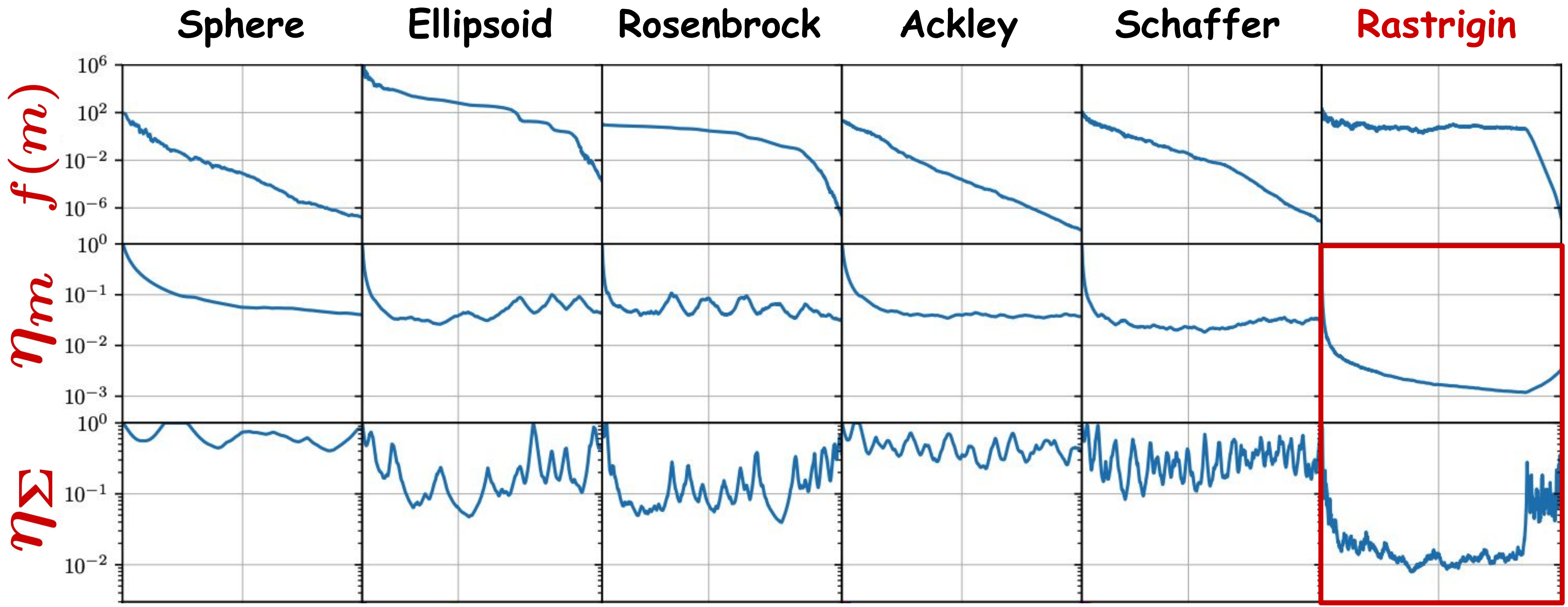


# Typical LRA-CMA behavior on 10-D noiseless problems




$\eta_m$ was slightly smaller on multimodal problems than on unimodal problems

# Typical LRA-CMA behavior on 10-D noiseless problems




$\eta_m$ and $\eta_\Sigma$ clearly decreased at the beginning

⇒ learning rates are adapted according to difficulty of search situations

# Typical LRA-CMA behavior on 10-D noisy problems

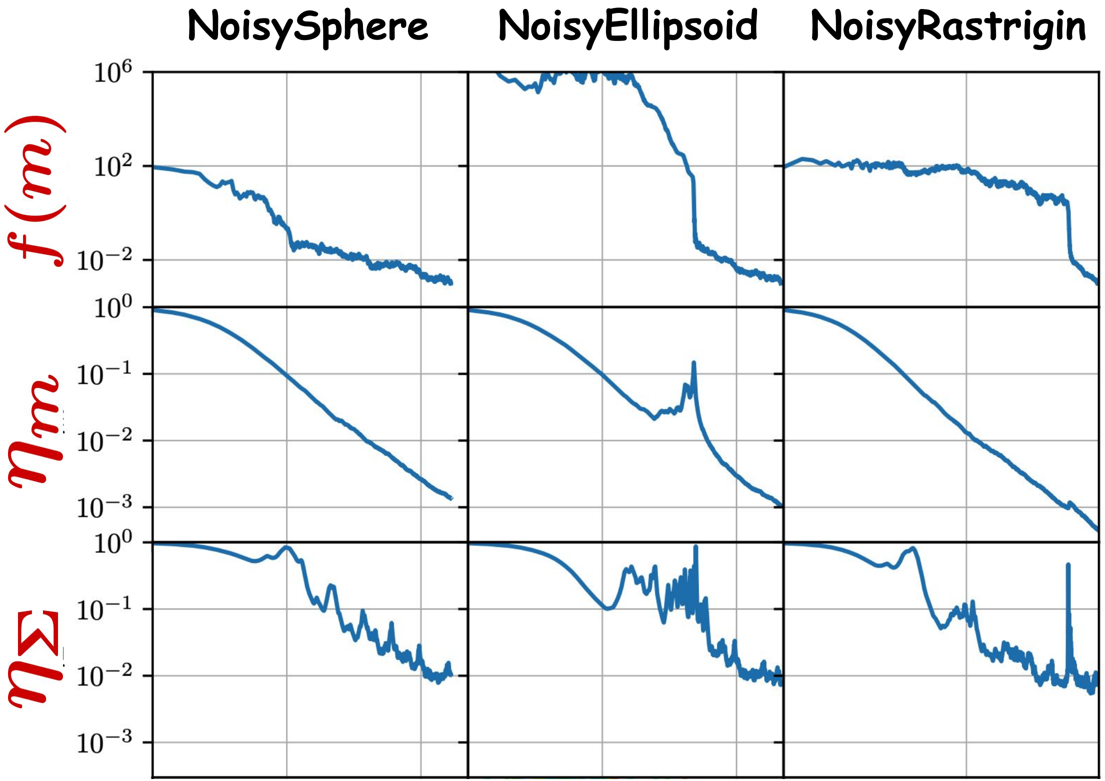
- Noise:
  - Early: negligible

behavior is similar to noiseless

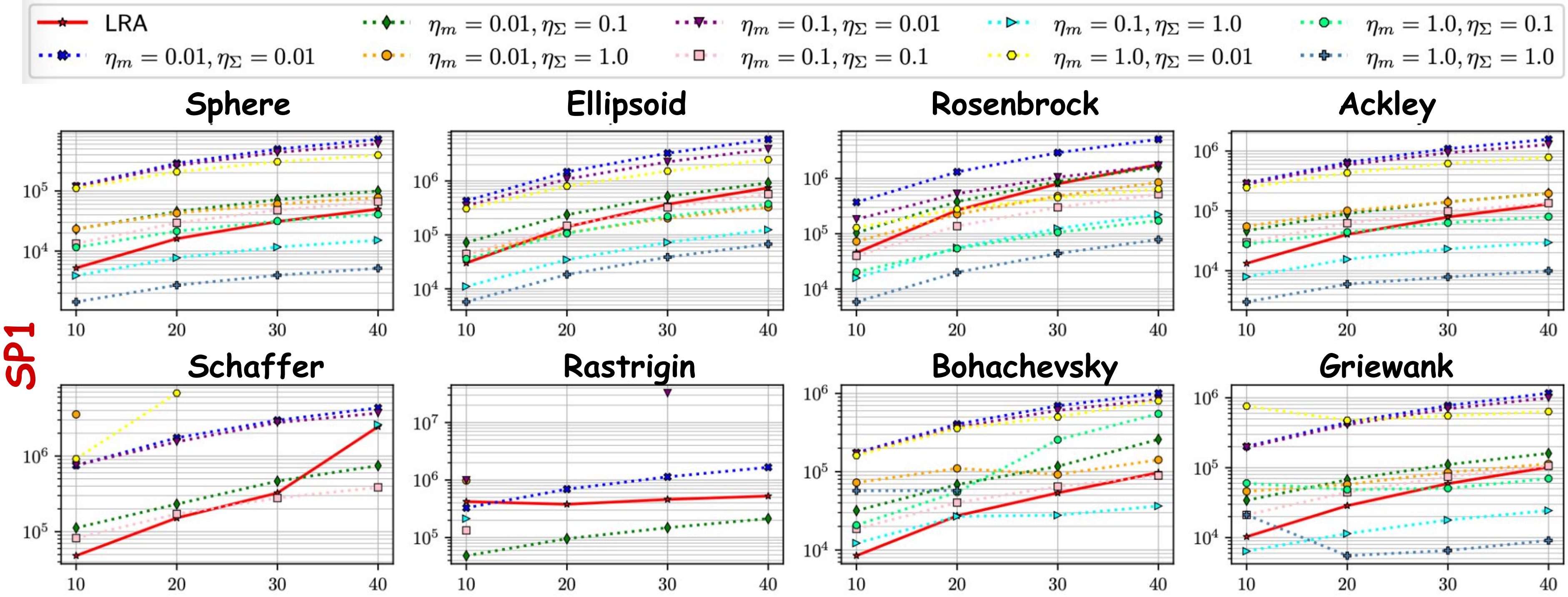
  - After: critical

fvalue approached noise scale

    - $\eta$ decreased



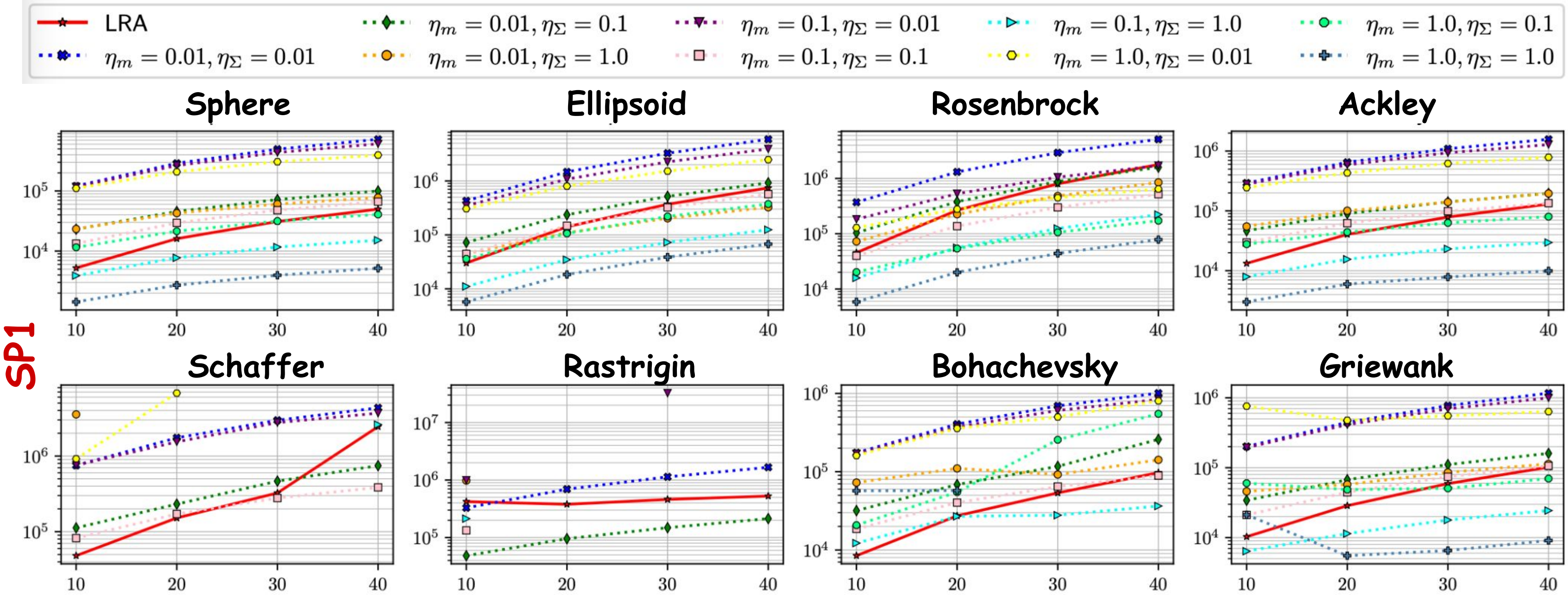
# SP1 versus (10-40)Dim. (Noiseless Problems)




with high $\eta \Rightarrow$ worse on multimodal, with small $\eta \Rightarrow$ slow on unimodal

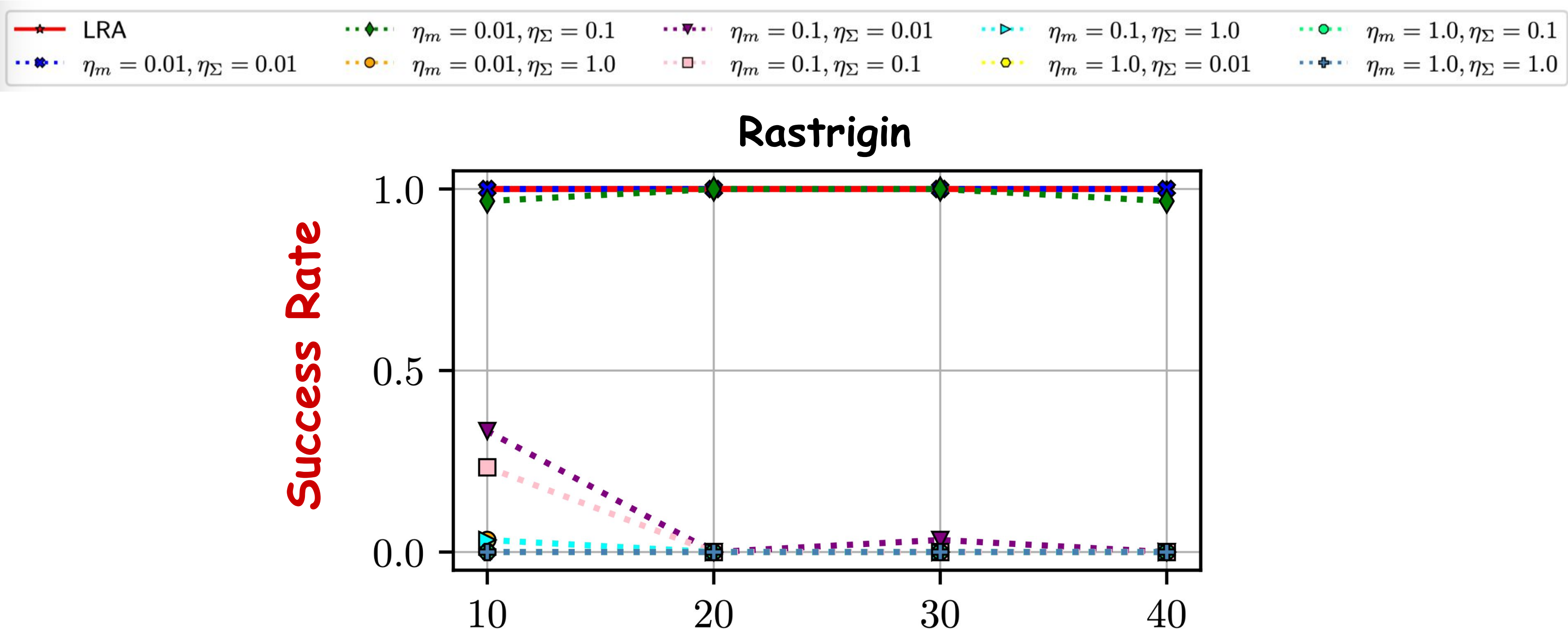
Clear trade-off in efficiency exists depending on $\eta$

# SP1 versus (10-40)Dim. (Noiseless Problems)



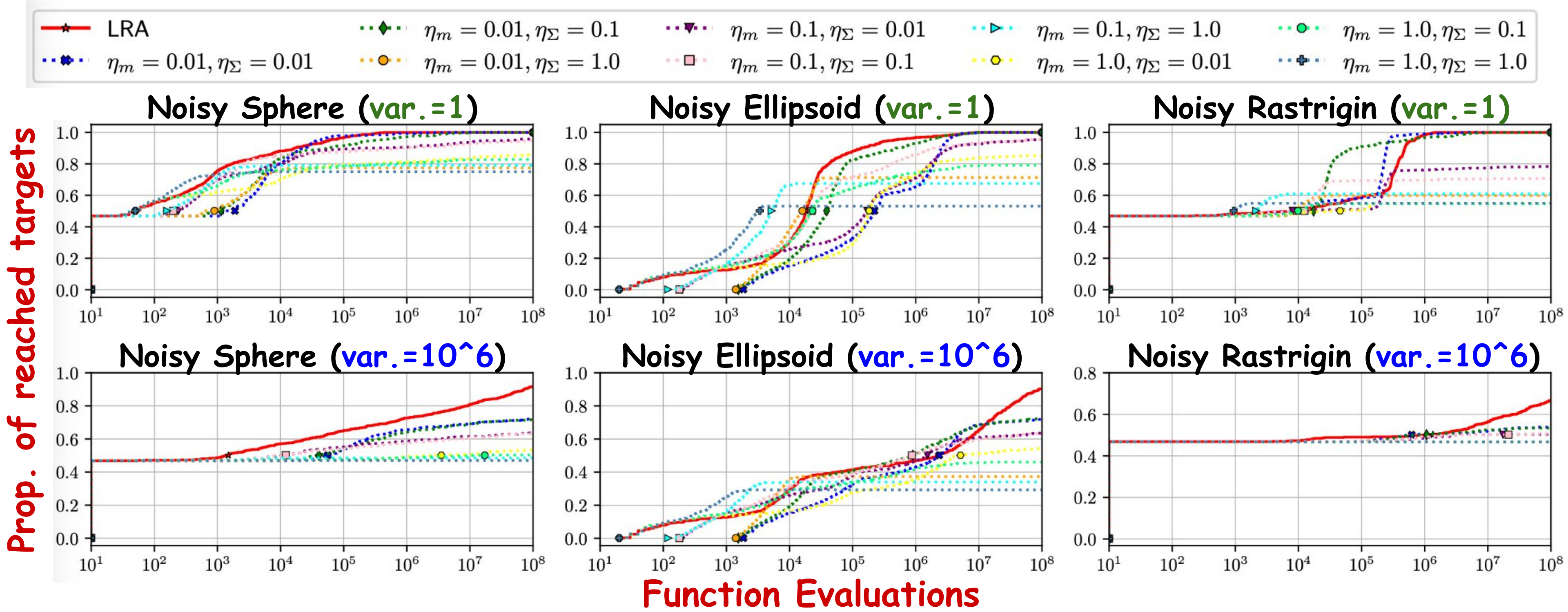

LRA shows stable and relatively good performance <u>without expensive tuning</u>

# Success Rate on Rastrigin Function




LRA with default λ (e.g. λ=15 for d=40) succeeded in all(n=30) trials on Rastrigin

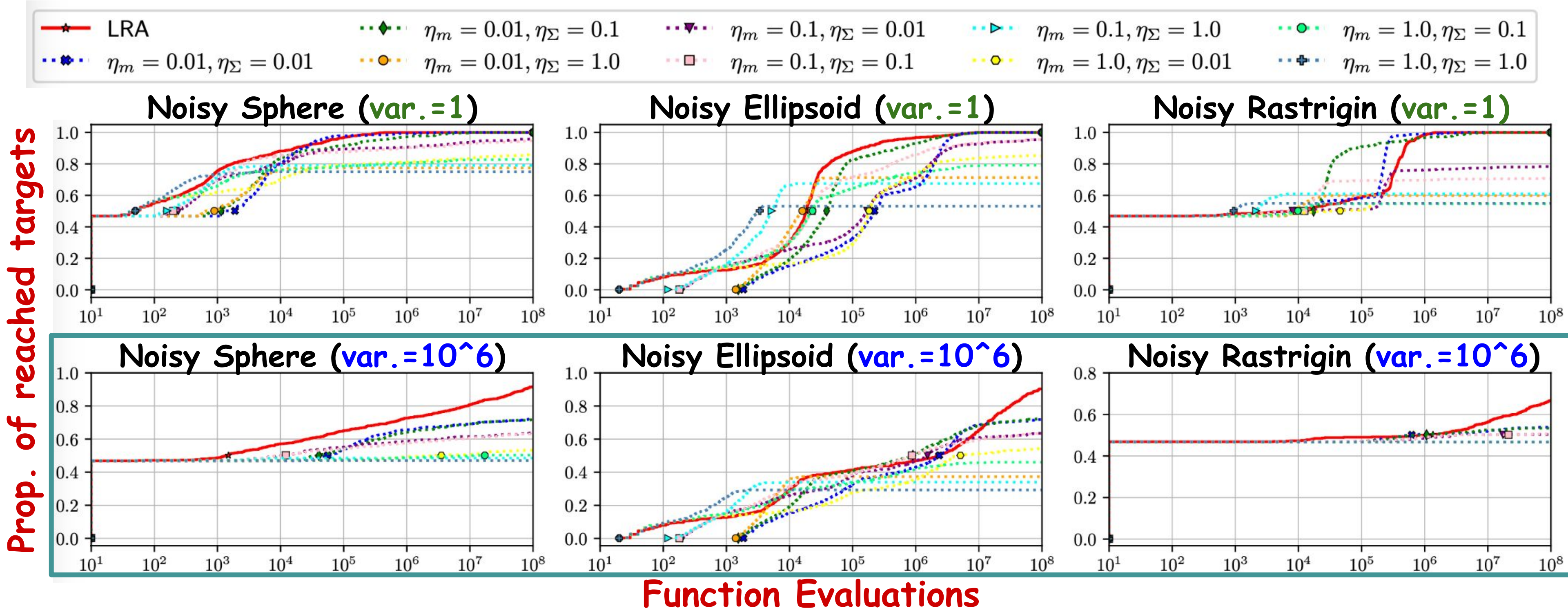
# Empirical cumulative density function on noisy problems




CMA with fixed $\eta$ had stopped improving the function value

In contrast, <u>*LRA continued to improve*</u> it even in strong noise

# Empirical cumulative density function on noisy problems




CMA with fixed $\eta$ had stopped improving the function value
In contrast, LRA continued to improve it even in strong noise

# Use of LRA-CMA with Python

- Available from CyberAgentAILab/cmaes (#star=233)

```
optimizer = CMA(mean=np.ones(10) * 3, sigma=2.0, lr_adapt=True)
```

Please create issues if you have any problems!

- (Scheduled for next month) Available from optuna (#star=8.4k)

private communication w/ optuna team

  - optuna:
    - popular BBO/HPO software (300k downloads/week)

***A wider audience can use LRA-CMA!***

# Conclusion

- Ultimate goal:
  - Hyperparameter-Free CMA-ES even for difficult problems
- Approach: LRA-CMA
  - Learning rate is adapted to maintain a positive constant SNR
- Evaluation:
  - LRA-CMA with default population size works well without tuning
- Future work:
  - Comparison against population size adaptation

Please feel free to contact masahironomura5325@gmail.com
if you have any questions!

# References & Appendix

# References


[Han16] Nikolaus Hansen. The CMA Evolution Strategy: A Tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

[HMK03] Nikolaus Hansen, Sibylle D M¨uller, and Petros Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary computation*, 11(1):1–18, 2003.

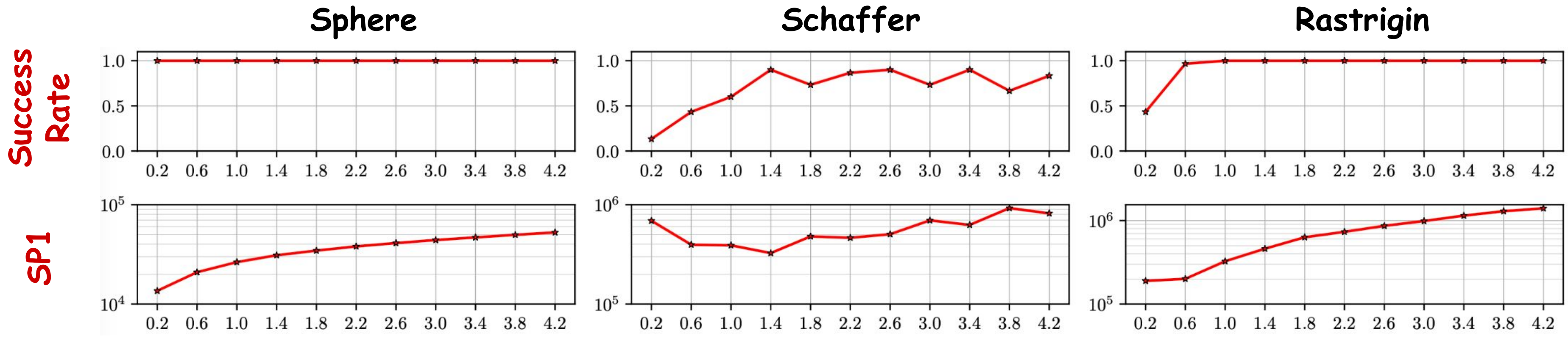
[MA17] Hidekazu Miyazawa and Youhei Akimoto. Effect of the Mean Vector Learning Rate in CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 721–728, 2017.

[NA16] Kouhei Nishida and Youhei Akimoto. Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient. In *Proceedings of the Genetic and Evolutionary Computation*, page 237–244, 2016.

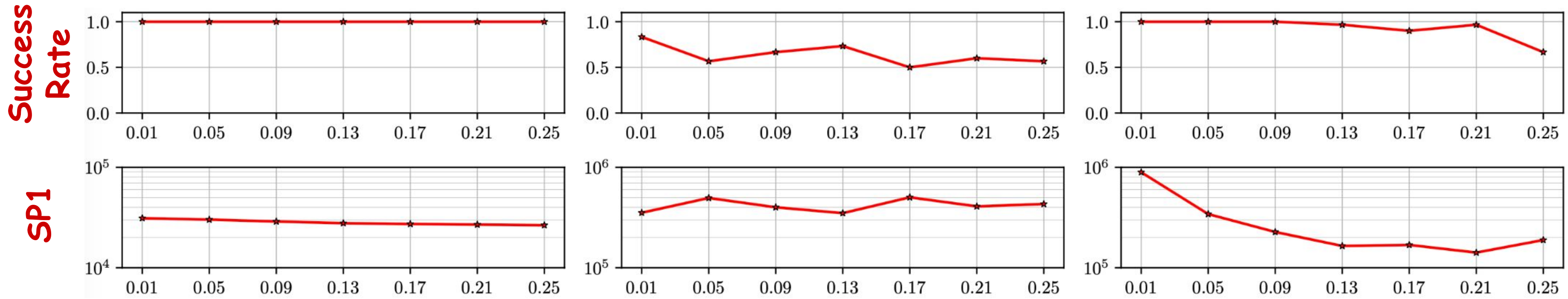
[NA18] Kouhei Nishida and Youhei Akimoto. PSA-CMA-ES: CMA-ES with Population Size Adaptation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, page 865–872, 2018.

# Appendix: Effects of Hyperparameters (1)

## α effect: success rate and SP1 on 30-D noiseless problems (30 trials)

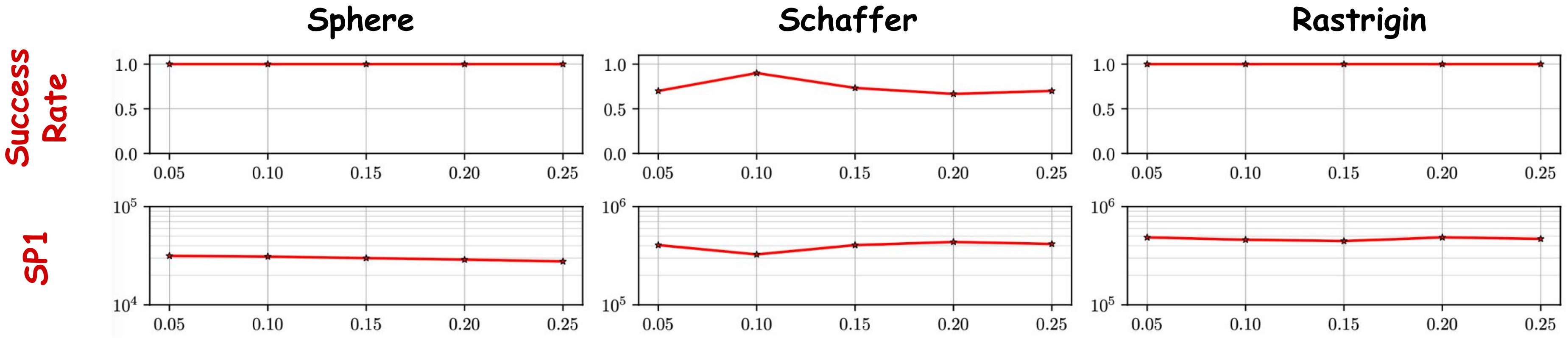



## $\beta_\Sigma$ effect: success rate and SP1 on 30-D noiseless problems (30 trials)

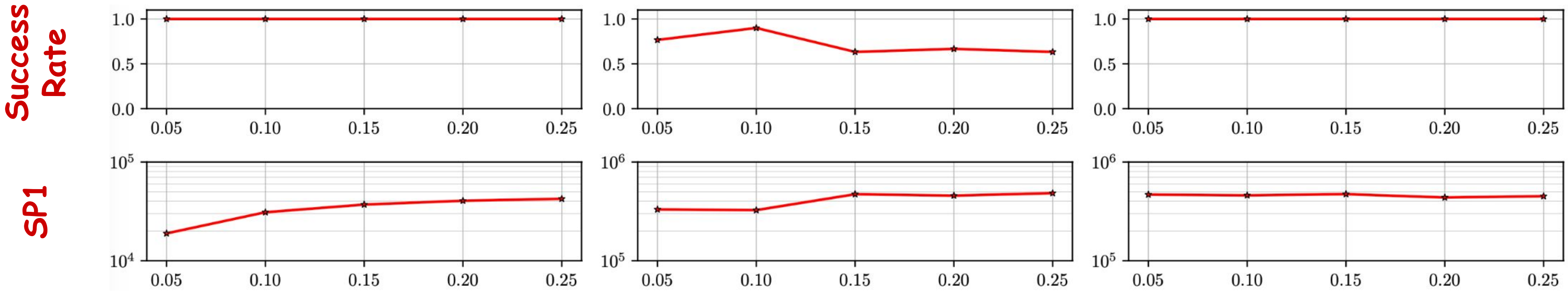


# Appendix: Effects of Hyperparameters (2)

## $\beta_m$ effect: success rate and SP1 on 30-D noiseless problems (30 trials)

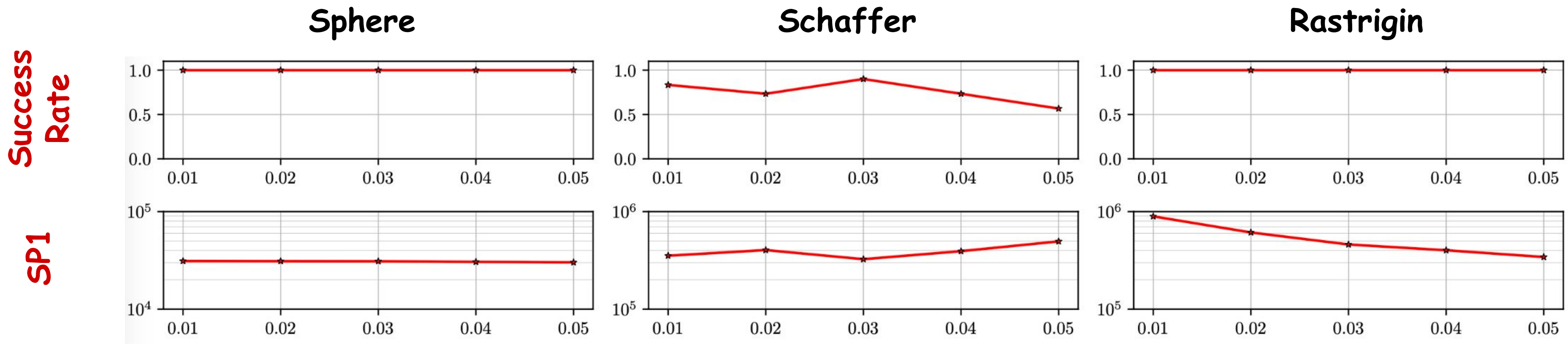



## γ effect: success rate and SP1 on 30-D noiseless problems (30 trials)



# Appendix: Effects of Hyperparameters (3)

$\beta_\Sigma$ effect (refined): success rate and SP1 on 30-D noiseless problems (30 trials)

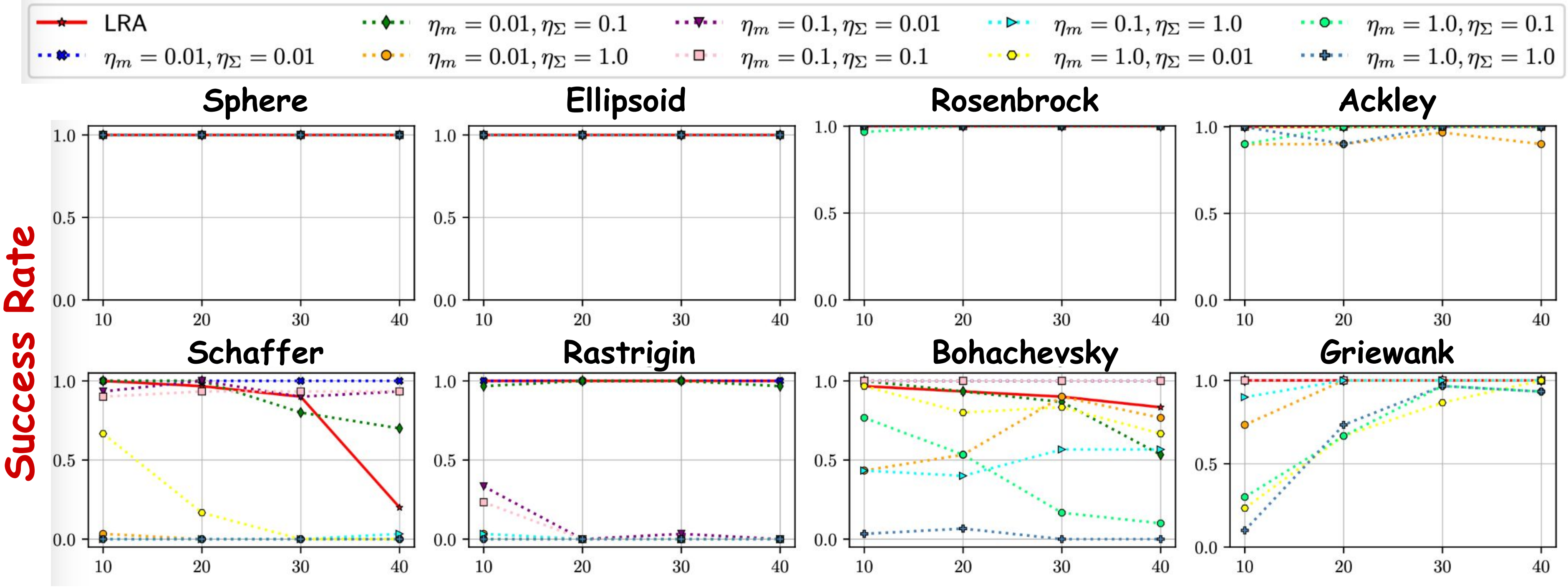


# Appendix: Benchmark Problems and Initial Distributions

| Definitions | Initial Distributions |
|---|---|
| $f_{\text{Sphere}}(x) = \sum_{i=1}^{d} x_i^2$ | $m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$ |
| $f_{\text{Ellipsoid}}(x) = \sum_{i=1}^{d} (1000^{\frac{i-1}{d-1}} x_i)^2$ | $m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$ |
| $f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{d-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ | $m^{(0)} = [0, \ldots, 0], \sigma^{(0)} = 0.1$ |
| $f_{\text{Ackley}}(x) = 20 - 20 \cdot \exp(-0.2\sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2}) + e - \exp(\frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x_i))$ | $m^{(0)} = [15.5, \ldots, 15.5], \sigma^{(0)} = 14.5$ |
| $f_{\text{Schaffer}}(x) = \sum_{i=1}^{d-1} (x_i^2 + x_{i+1}^2)^{0.25} \cdot [\sin^2(50 \cdot (x_i^2 + x_{i+1}^2)^{0.1}) + 1]$ | $m^{(0)} = [55, \ldots, 55], \sigma^{(0)} = 45$ |
| $f_{\text{Rastrigin}}(x) = 10d + \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i))$ | $m^{(0)} = [3, \ldots, 3], \sigma^{(0)} = 2$ |
| $f_{\text{Bohachevsky}}(x) = \sum_{i=1}^{d-1} (x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$ | $m^{(0)} = [8, \ldots, 8], \sigma^{(0)} = 7$ |
| $f_{\text{Griewank}}(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \Pi_{i=1}^{d} \cos(x_i/\sqrt{i}) + 1$ | $m^{(0)} = [305, \ldots, 305], \sigma^{(0)} = 295$ |

Although the Rosenbrock function has local minima,
in our setting, it could be regarded as an almost unimodal problem

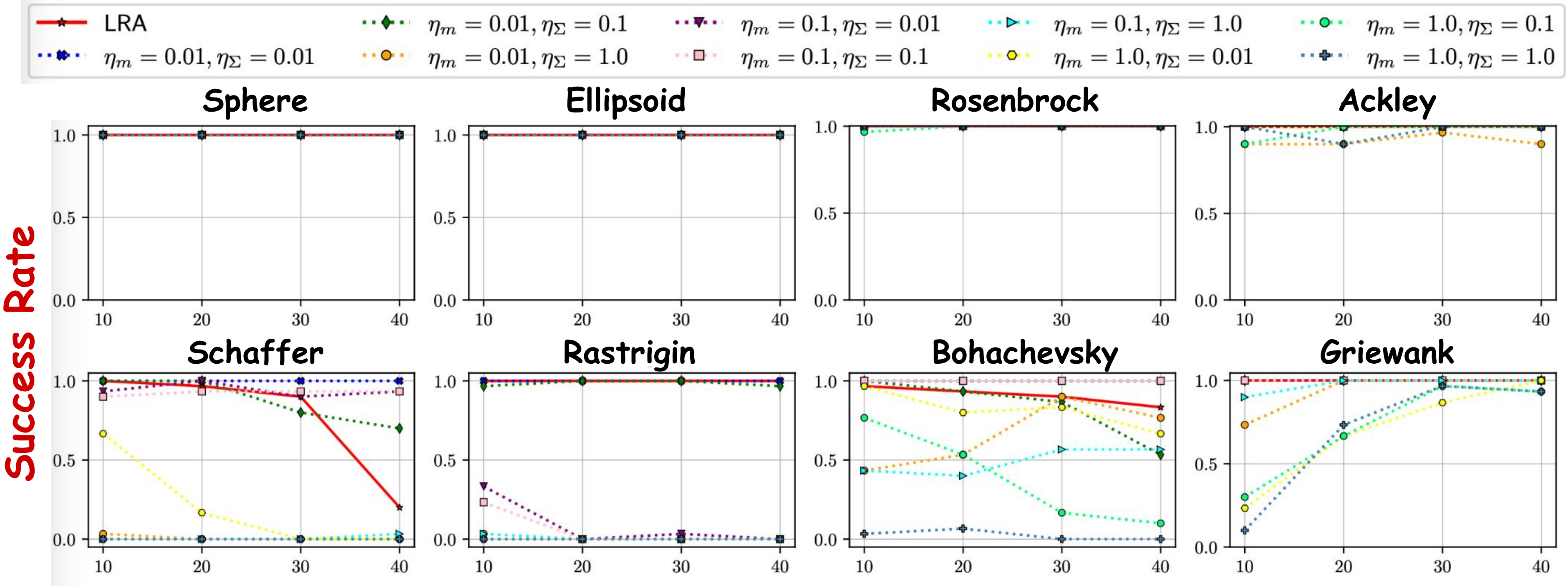
# Success Rate versus (10-40)Dim. (Noiseless Problems)




For multimodal, CMA with high $\eta$ often *failed*, but with small $\eta$ had a high SR

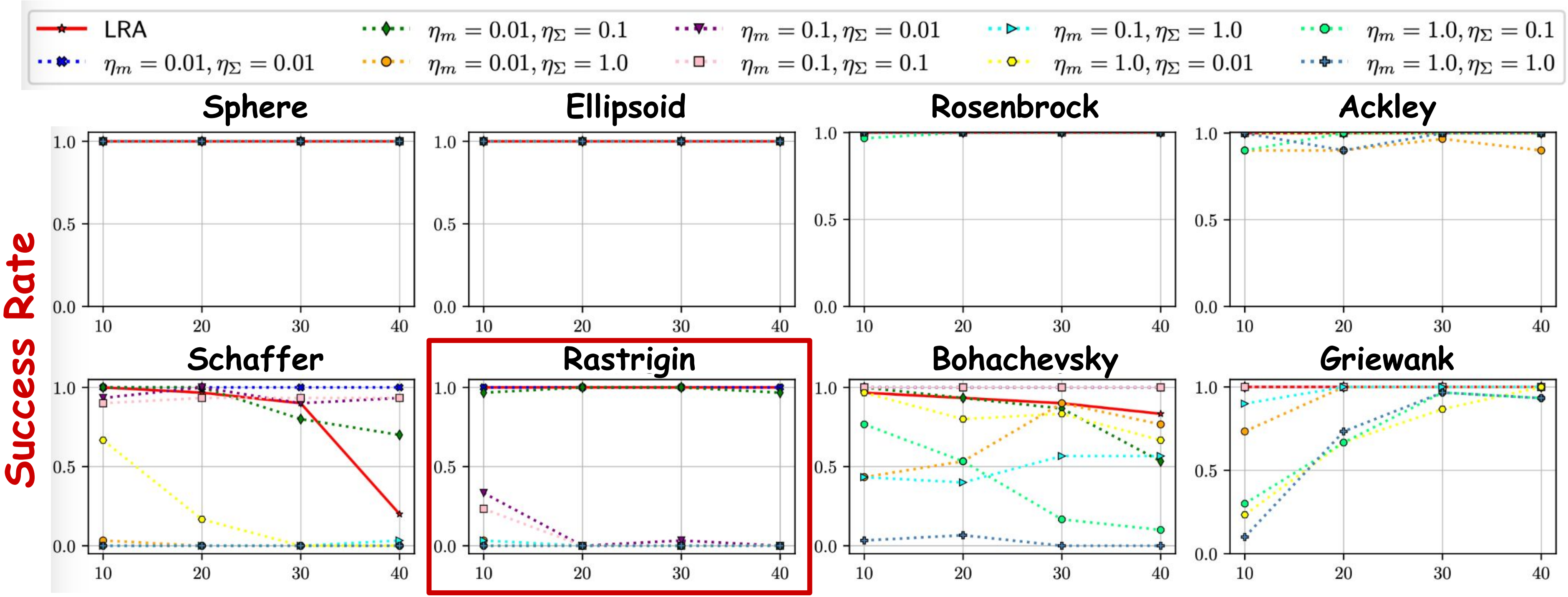
Success is highly dependent on the η setting

# Success Rate versus (10-40)Dim. (Noiseless Problems)



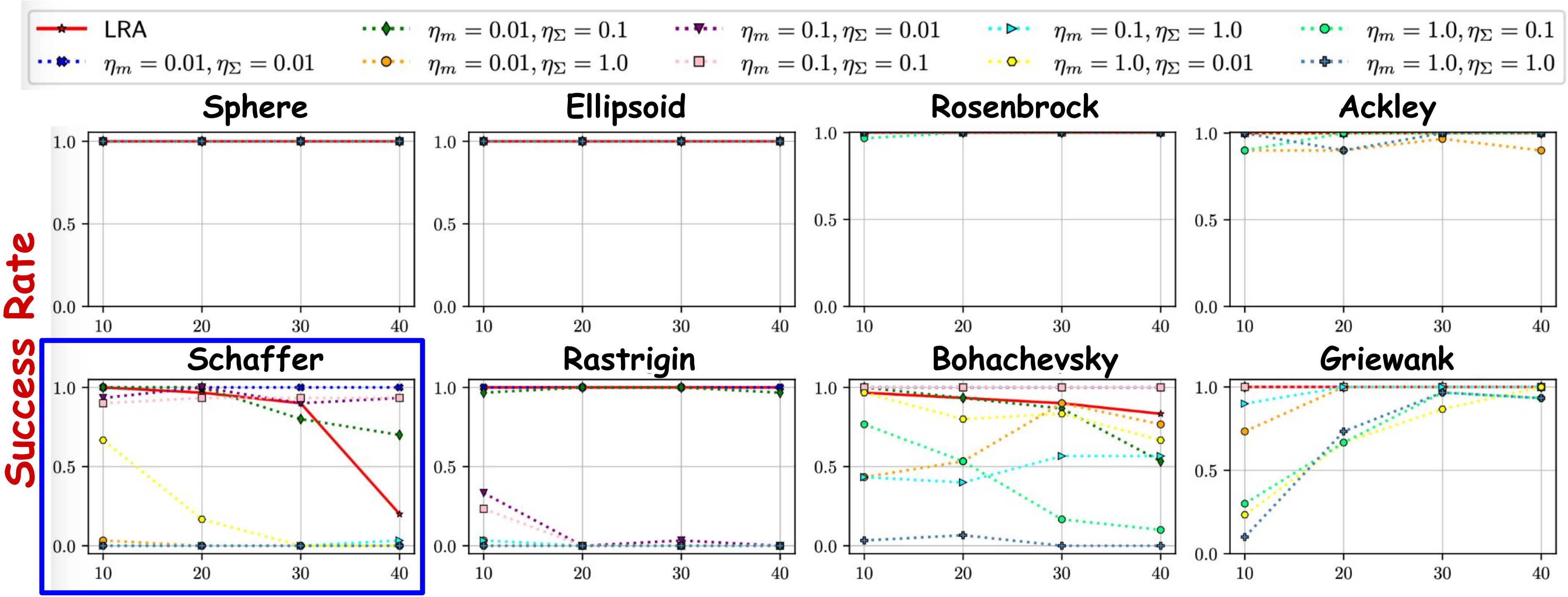

LRA-CMA had a relatively *good Success Rate without η tuning*

# Success Rate versus (10-40)Dim. (Noiseless Problems)




<u>*LRA with default λ (e.g. λ=15 for d=40) succeeded in all trials on Rastrigin*</u>

# Success Rate versus (10-40)Dim. (Noiseless Problems)




LRA performance degrades on Schaffer with d=40 ⇒ future work

# Appendix: Step-Size Correction

- When the learning rate for *m* is updated, appropriate step-size changes

- Quality gain analysis for optimal step-size:

$$\sigma^* \propto 1/\eta_m$$

on (infinite-dim) convex quadratic functions

- *To maintain the optimal step-size*, we perform step-size correction:

$$\sigma^{(t+1)} \leftarrow \frac{\eta_m^{(t)}}{\eta_m^{(t+1)}} \sigma^{(t+1)}$$