Covariance Matrix Adaptation - Evolution Strategy: A summary

Gaëtan Serré
ENS Paris-Saclay - Centre Borelli
gaetan.serre@ens-paris-saclay.fr

1. Introduction

The *Covariance Matrix Adaptation - Evolution Strategy* (CMA-ES) is a global and black-box optimization algorithm. It is a randomized black-box search algorithm as it samples evaluation points from a distribution conditionned with the previous parameters. This kind of algorithm is detailed in Algorithm 1. CMA-ES uses a multivariate Gaussian as the sampling distribution $\mathcal{N}(m, C)$. The authors made this choice as, given the variances and covariances between components, the normal distribution has the largest entropy in $\mathbb{R}^d$. To goal is to find how update the mean and covariance matrix of this distribution to minimize the trade-off between finding a good approximation of the optimum and evaluate the objective function as few times as possible. In this small paper, we will present the ideas behind CMA-ES. For more details, see (Hansen 2023). Throughout this paper, we will suppose that the objective function is to be maximized.

For g in $1...k$:

1. *Let d_θ a distribution on $\mathcal{X}$ parametrized by θ;*
2. *Sample λ points: $(x_i)_{1\leq i\leq\lambda} \sim d_{\theta_i}$;*
3. *Evaluate the points: $f\Big((x_i)_{1\leq i\leq\lambda}\Big)$;*
4. *Update the parameters $\theta_{i+1} = F(\theta_i, (x_1, f(x_1)), ..., (x_\lambda, f(x_\lambda)))$.*

Algorithm 1: Black-box search algorithm.

2. Update the mean

In the CMA Evolution Strategy, the λ points are sampled from a multivariate Gaussian distribution which writes:

$$\Big(x_i^{(g)}\Big)_{1\leq i\leq\lambda} \sim m^{(g)} + \sigma^{(g)}\mathcal{N}\Big(0, C^{(g)}\Big), \tag{1}$$

where g is the generation number, $m^{(g)}$ is the mean vector, $\sigma^{(g)}$ is the "overall" standard deviation and $C^{(g)}$ is the covariance matrix. It is equivalent to say that $x_i^{(g)} \sim \mathcal{N}\Big(m^{(g)}, \big(\sigma^{(g)}\big)^2 C^{(g)}\Big)$. To update the mean, we begin by selecting the μ best points, i.e.:

$$f\Big(x_1^{(g)}\Big) \geq ... \geq f\Big(x_\mu^{(g)}\Big) \geq f\Big(x_{\mu+1}^{(g)}\Big) \geq ... f\Big(x_\lambda^{(g)}\Big). \tag{2}$$

We introduce the index notation $i : \lambda$, denoting the index of the i-th best point. The mean at generation $g + 1$ becomes a weighted average of those points:

$$m^{(g+1)} = m^{(g)} + c_m \sum_{i=1}^{\mu} w_i \big(x_{i:\lambda} - m^{(g)}\big), \tag{3}$$

where:

$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq ... \geq w_\mu \geq 0, \tag{4}$$

and c_m is a learning rate, usually set to 1. In that case, Eq. 3 simply becomes:

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}. \tag{5}$$

The choice of the weights is crucial in CMA-ES as they represent the trade-off between exploration and exploitation. To do so, we define the quantity μ_{eff} as:

$$\mu_{\text{eff}} = \left(\frac{\|w\|_1}{\|w\|_2}\right)^2 = \frac{\left(\sum_{i=1}^{\mu} |w_i|\right)^2}{\sum_{i=1}^{\mu} w_i^2} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}. \tag{6}$$

From Eq. 4, one can easily derive $1 \leq \mu_{\text{eff}} \leq \mu$, the latter happens when all the weights are equal, i.e. $\forall 1 \leq i \leq \mu, w_i = \frac{1}{\mu}$. μ_{eff} quantize the loss of variance due to the selection of the best points. According to the author, $\mu_{\text{eff}} \approx \frac{\lambda}{4}$ indicates a reasonable choice of w_i. A simple and decent way to achieve that is to set $w_i \propto \mu - i + 1$ (see 5.2). Choosing $c_m < 1$ can work well on noisy function. However, the step-size σ is roughly proportional to $\frac{1}{c_m}$ and thus, with a too small c_m, the search would moves away from the current region of relevance.

3. Update the covariance matrix

To update the covariance matrix, we need to estimate it using the points $(x_i)_{1 \leq i \leq \lambda}$. In this section, we assume $\sigma = 1$ for simplicity. If $\sigma \neq 1$, one can simply rescale the covariance matrix by $\frac{1}{\sigma^2}$. If we have enough sample, one can use the empirical covariance matrix:

$$C_{\text{emp}}^{(g+1)} = \frac{1}{\lambda - 1} \sum_{i=1}^{\lambda} \left(x_i^{(g+1)} - \frac{1}{\lambda} \sum_{i=1}^{\lambda} x_i^{(g+1)}\right) \left(x_i^{(g+1)} - \frac{1}{\lambda} \sum_{i=1}^{\lambda} x_i^{(g+1)}\right)^\top. \tag{7}$$

A different would be to use the real mean $m^{(g+1)}$ computed before instead of the empirical mean:

$$C_{\lambda}^{(g+1)} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \Big(x_i^{(g+1)} - m^{(g+1)}\Big) \Big(x_i^{(g+1)} - m^{(g+1)}\Big)^\top. \tag{8}$$

Both are unbiaised estimators of the covariance matrix. However, they do not influence the search towards the direction of the μ best points. To do so, one can use the same *weighted selection* as in Eq. 3:

$$C_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} w_i \Big(x_{i:\lambda}^{(g+1)} - m^{(g)}\Big) \Big(x_{i:\lambda}^{(g+1)} - m^{(g)}\Big)^\top. \tag{9}$$

This last estimator tends to reproduce the current best points and thus allows a faster convergence. However, this estimation method requires a lot of samples for μ_{eff} must be large

enough to be reliable. The author suggests another method to estimate $C^{(g+1)}$ that tackles these two issues, the *rank-μ* method.

3.1. Rank-μ method

As stated before, for Eq. 9 to be a reliable estimator, one need a lot of sample, as ideally $\mu_{\text{eff}} \approx \frac{\lambda}{4}$. However, evaluate the function is often the main bottleneck of the algorithm. The author suggests to use the *rank-μ* method to estimate the covariance matrix. The idea behind this method is to use information of previous generations in the estimation of the next one:

$$C^{(g+1)} = \frac{1}{g+1} \sum_{i=0}^{g} \frac{1}{{\sigma^{(i)}}^2} C_\mu^{(i+1)}, \tag{10}$$

where $\sigma^{(i)}$ is the step-size at generation i. In Eq. 10, each generation has the same weight in the estimation of the covariance matrix of the next generation. A natural idea would be to give more weight to the most recent generations through exponential smoothing:

$$C^{(g+1)} = \Big(1 - c_\mu\Big) C^{(g)} + c_\mu \frac{1}{{\sigma^{(g)}}^2} C_\mu^{(g+1)}, \tag{11}$$

where $c_\mu \leq 1$ is a learning rate. The author suggests that $c_\mu \approx \min\Big(1, \frac{\mu_{\text{eff}}}{d^2}\Big)$ is a reasonable choice. Eq. 11 can be written as:

$$C^{(g+1)} = \Big(1 - c_\mu\Big) C^{(g)} + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)} {y_{i:\lambda}^{(g+1)}}^\top, \tag{12}$$

where $y_{i:\lambda}^{(g+1)} = \frac{x_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$. This method is so called *rank-μ* as the rank of the sum of the dot products is $\min(\mu, d)$. Finally, the author generalizes Eq. 12 with λ weights that does not requires to sum to 1 nor being positive:

$$C^{(g+1)} = \left(1 - \sum_{i=1}^{\lambda} w_i c_\mu\right) C^{(g)} + c_\mu \sum_{i=1}^{\lambda} w_i y_{i:\lambda}^{(g+1)} {y_{i:\lambda}^{(g+1)}}^\top. \tag{13}$$

Usually, $\sum_{i=1}^{\mu} w_i = 1 = -\sum_{i=\mu+1}^{\lambda} w_i$. To emphasize the importance of c_μ, the author introduce the *backward time horizon* Δg. It represent the number of generations used to encode roughly 63% of the information of the estimation of the covariance matrix of the next generation. E.g. if $\Delta g = 10$, it means that the 10 last generations are used to compute 63% of the information of the covariance matrix of the next generation. Indeed, Eq. 11 can be extended to:

$$C^{(g+1)} = \Big(1 - c_\mu\Big)^{(g+1)} C^{(0)} + c_\mu \sum_{i=0}^{g} \Big(1 - c_\mu\Big)^{g-i} \frac{1}{{\sigma^{(i)}}^2} C_\mu^{(i+1)}. \tag{14}$$

Therefore, Δg is defined by:

$$c_\mu \sum_{i=g+1-\Delta g}^{g} \Big(1 - c_\mu\Big) \approx 0.63 \approx 1 - \frac{1}{e}. \tag{15}$$

One can solve this equation to find $\Delta g \approx \frac{1}{c_\mu}$ (see 5.3). It shows that, the smaller is c_μ the more past generations are used to compute the covariance matrix.

3.2. Rank-1 method

In the previous section, we defined a method that uses information of the entire population to update the covariance matrix. The author present the *rank-1* method that uses only the best point of the population to update the covariance matrix. This method highlights the correlation between generations and the final update rule of CMA-ES uses both *rank-μ* and *rank-1* methods. The idea of the *rank-1* method is simply to use only the best point to update the covariance matrix in order to increase the likelihood of reproducing this point in the next generation. The update rule is just an adaption of Eq. 12 where only $y_{1:\lambda}^{(g+1)}$ is used:

$$C^{(g+1)} = \left(1 - \sum_{i=1}^{\lambda} w_i c_1\right) C^{(g)} + c_1 y_{1:\lambda}^{(g+1)} y_{1:\lambda}^{(g+1)^\top}, \tag{16}$$

where, according to the author, $c_1 \approx \frac{2}{n^2}$ is a good choice. This method however, discards the sign information of $y_{1:\lambda}^{(g+1)}$ as

$$y_{1:\lambda}^{(g+1)} y_{1:\lambda}^{(g+1)^\top} = -y_{1:\lambda}^{(g+1)} \left(-y_{1:\lambda}^{(g+1)^\top}\right). \tag{17}$$

To reintroduce this information in the estimation of the covariance matrix, the author suggests to build a *evolution path* $p_c^{(g)}$:

$$p_c^{(g+1)} = (1 - c_c) p_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \frac{\sum_{i=1}^{\mu} w_i \left(x_i^{(g+1)} - m^{(g)}\right)}{\sigma^{(g)}}. \tag{18}$$

This represent a exponential smoothing of the sum:

$$\sum_g \frac{\sum_{i=1}^{\mu} w_i \left(x_i^{(g+1)} - m^{(g)}\right)}{\sigma^{(g)}}. \tag{19}$$

We set $p_c^{(0)} = 0$ and $\sqrt{c_c(2 - c_c)\mu_{\text{eff}}}$ is a normalization constant in order to $p_c^{(g)} \sim y_{1:\lambda}^{(g)} \sim \mathcal{N}\left(0, C^{(g)}\right)$ (see 5.4). The final update rule of the covariance matrix of CMA-ES combine Eq. 12 and Eq. 19:

$$C^{(g+1)} = \left(\underbrace{1 - c_1 - c_\mu \sum_{i=1}^{\lambda} w_i}_{\approx 0}\right) C^{(g)} + c_1 \underbrace{p_c^{(g+1)} p_c^{((g+1))^\top}}_{\text{rank-1 update}} + \underbrace{c_\mu \sum_{i=1}^{\lambda} w_i y_{i:\lambda}^{(g+1)} y_{i:\lambda}^{(g+1)^\top}}_{\text{rank-}\mu\text{ update}}, \tag{20}$$

where

- $c_1 \approx \frac{2}{n^2}$,
- $c_\mu \approx \min\left(1 - c_1, \frac{\mu_{\text{eff}}}{n^2}\right)$,
- $y_{i:\lambda}^{(g+1)} = \frac{x_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$,
- $\sum_{i=1}^{\lambda} w_i \approx -\frac{c_1}{c_\mu}$.

4. Choice of the step-size

The last parameter to choose is the step-size $\sigma^{(g)}$. The author suggests to control the step-size using the evolution of the direction of the steps, represented by $y_{i:\lambda}^{(g)}$. One can dinstinguish three cases:

1. the steps goes in the same direction, meaning that the optimum is likely to be in this direction and increasing the step-size allows to reach the optimum faster;
2. the steps cancel each other, meaning the the optimum is likely to be in the interior of the region "drawn" by the steps, and decreasing the step-size allows to explore that region;
3. the steps follows almost orthogonal directions, meaning that the algorithm follows the contour lines of the objective function and the step-size must be kept constant.

Theses cases are illustrated in Figure 1. One way to infer in which case steps are is to use the length of the evolution path $p_\sigma^{(g)}$:

$$p_\sigma^{(g)} = \sum_g \frac{\sum_{i=1}^{\mu} w_i \left(x_i^{(g)} - m^{(g-1)} \right)}{\sigma^{(g-1)}}. \tag{21}$$

As before, the author used an exponential smoothing to compute $p_\sigma^{(g)}$:

$$p_\sigma^{(g+1)} = (1 - c_\sigma) p_\sigma^{(g)} + \sqrt{c_\sigma (2 - c_\sigma) \mu_{\text{eff}}} C^{-\frac{1}{2}} \frac{\sum_{i=1}^{\mu} w_i \left(x_i^{(g+1)} - m^{(g)} \right)}{\sigma^{(g)}}. \tag{22}$$

Using the same reasoning as in 5.4, one can show that, if $p_\sigma^{(0)} \sim \mathcal{N}(0, I)$, $p_\sigma^{(g)} \sim \mathcal{N}(0, I)$. This way, one can compare the length of the path represented by $\left\| p_\sigma^{(g)} \right\|$ to its expected value and state on how update $\sigma^{(g)}$. The author suggests to update $\sigma^{(g)}$ as follow:

$$\sigma^{g+1} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\left\| p_\sigma^{g+1} \right\|}{\mathbb{E}[\|\mathcal{N}(0, I)\|]} - 1 \right) \right), \tag{23}$$

where $d_\sigma \approx 1$ and $\mathbb{E}[\|\mathcal{N}(0, I)\|] \approx \sqrt{d}$. The author provide default values for all parameters of CMA-ES, summarized in Table 1.

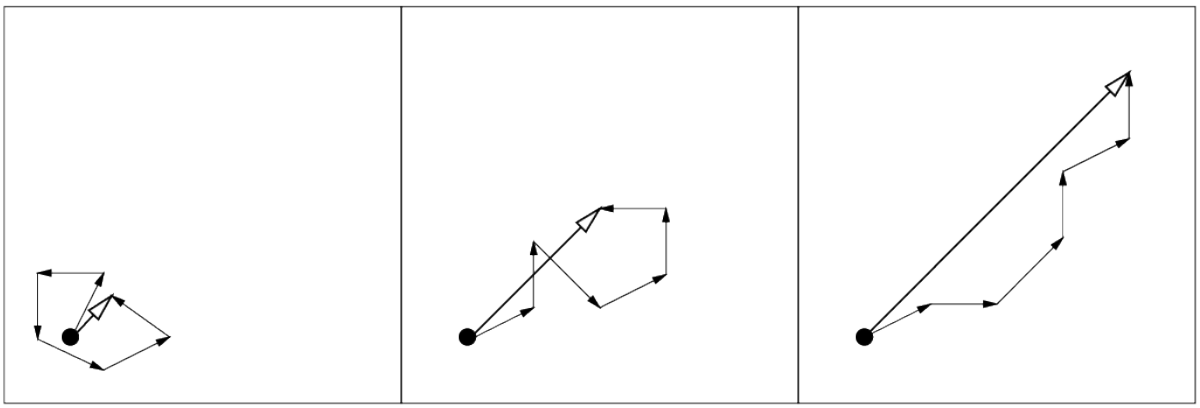

Figure 1: *Left*: Steps cancels each other, step-size must be decreased. *Right*: Steps are correlated and goes to the same direction, step-size must be increased. *Middle*: Steps follows almost orthogonal directions, step-size must be kept constant.

Bibliography

N. Hansen, "The CMA Evolution Strategy: A Tutorial," arXiv, 2023. Accessed: Apr. 4, 2023. [Online]. Available: http://arxiv.org/abs/1604.00772 (Comment: ArXiv e-prints, arXiv:1604.00772, 2016, pp.1-39)

5. Annex

5.1. Default parameters

Parameters	Default value
λ	$4 + \lfloor 3 \ln d \rfloor$
μ	$\left\lceil \frac{\lambda}{2} \right\rceil$
$w_i{}'$	$\ln \frac{\lambda+1}{2} - \ln i,\ \forall i \in [\![1 ... \lambda]\!]$
μ_{eff}^-	$\frac{\left(\sum_{i=\mu+1}^{\lambda} w_i{}'\right)^2}{\sum_{i=\mu+1}^{\lambda} w_i{}'^2}$
α_{cov}	2
c_μ	$\min\left(1 - c_1, \alpha_{\text{cov}} \frac{\frac{1}{4} + \mu_{\text{eff}} + \frac{1}{\mu_{\text{eff}}} - 2}{(n+2)^2 + \alpha_{\text{cov}} \frac{\mu_{\text{eff}}}{2}}\right)$
c_1	$\frac{\alpha_{\text{cov}}}{(n+1.3)^2 + \mu_{\text{eff}}}$
c_c	$\frac{4 + \frac{\mu_{\text{eff}}}{n}}{n+4+2\frac{\mu_{\text{eff}}}{n}}$
α_μ^-	$1 + \frac{c_1}{c_\mu}$
$\alpha_{\mu_{\text{eff}}}^-$	$1 + \frac{2\mu_{\text{eff}}^-}{\mu_{\text{eff}}+2}$
$\alpha_{\text{pos def}}^-$	$\frac{1-c_1-c_\mu}{d} c_\mu$
w_i	$\begin{cases} \frac{1}{\sum_j \lvert w_j{}' \rvert^+} w_i{}' & \text{if } w_i{}' \geq 0 \text{ (sum to 1)} \\ \left(\frac{\min\left(\alpha_\mu^-, \alpha_{\mu_{\text{eff}}}^-, \alpha_{\text{pos def}}^-\right)}{\sum_j \lvert w_j{}' \rvert^-}\right) & \text{if } w_i{}' < 0 \end{cases}$
c_σ	$\frac{\mu_{\text{eff}}+2}{d+\mu_{\text{eff}}+5}$
d_σ	$1 + 2\max\left(0, \sqrt{\frac{\mu_{\text{eff}}-1}{d+1}} - 1\right)$

Table 1: Default parameters of CMA-ES.

5.2. μ_{eff}

Proof. Let $w_i = \frac{\mu - i + 1}{\sum_{i=1}^{\mu} \mu - i + 1}$. Then:

$$
\begin{aligned}
\mu_{\text{eff}} &= \frac{1}{\sum_{i=1}^{\mu}\left(\frac{\mu-i+1}{\sum_{i=1}^{\mu}\mu-i+1}\right)^2} = \frac{1}{\sum_{i=1}^{\mu}\left(\frac{i}{\sum_{i=1}^{\mu}i}\right)^2} \\
&= \frac{1}{\sum_{i=1}^{\mu}\frac{i^2}{\left(\frac{\mu(\mu+1)}{2}\right)^2}} = \frac{1}{\frac{\mu(\mu+1)(2\mu+1)}{6\left(\frac{\mu(\mu+1)}{2}\right)^2}} \\
&= \frac{6\left(\frac{\mu(\mu+1)}{2}\right)^2}{\mu(\mu+1)(2\mu+1)} = \frac{3\mu(\mu^3+2\mu^2+\mu)}{2(2\mu^3+3\mu^2+\mu)} \\
&= \frac{3\mu(1+\mu)}{2(1+2\mu)} \approx \frac{\frac{3\lambda}{2}\left(1+\frac{\lambda}{2}\right)}{2(1+\lambda)} \\
&= \frac{\frac{3\lambda^2+6\lambda}{2}}{4(1+\lambda)} = \frac{3\lambda(2+\lambda)}{8(1+\lambda)} \\
&\approx \frac{3\lambda}{8}.
\end{aligned}
\tag{24}
$$

■

5.3. Δg

Proof.

$$
\begin{aligned}
c_\mu \sum_{i=g+1-\Delta g}^{g}\left(1-c_\mu\right)^{g-i} &= c_\mu\left(\left(1-c_\mu\right)^{\Delta g-1}+\left(1-c_\mu\right)^{\Delta g-2}+...+\left(1-c_\mu\right)^{0}\right) \\
&= c_\mu \sum_{i=0}^{\Delta g-1}\left(1-c_\mu\right)^{i} \\
&= c_\mu \frac{\left(1-c_\mu\right)^{0}-\left(1-c_\mu\right)^{\Delta g}}{1-\left(1-c_\mu\right)} \\
&= c_\mu \frac{1-\left(1-c_\mu\right)^{\Delta g}}{c_\mu} = 1-\left(1-c_\mu\right)^{\Delta g}.
\end{aligned}
\tag{25}
$$

Thus, the problem becomes to find Δg such that $1-\left(1-c_\mu\right)^{\Delta g} \approx 0.63 \approx 1-\frac{1}{e}$:

$$
\begin{aligned}
& 1-\left(1-c_\mu\right)^{\Delta g} = 1-\frac{1}{e} \\
\Leftrightarrow\ & \left(1-c_\mu\right)^{\Delta g} = e^{-1} \\
\Leftrightarrow\ & \Delta g \ln\left(1-c_\mu\right) = -1 \\
\Leftrightarrow\ & \Delta g = -\frac{1}{\ln\left(1-c_\mu\right)} \approx \frac{1}{c_\mu} \text{(using Taylor's expansion of order 1).}
\end{aligned}
\tag{26}
$$

■

5.4. $p_c^{(g)} \sim \mathcal{N}(0, C)$

Proof. Under the assumption that $p_c^{(g)} \sim \mathcal{N}\left(0, C^{(g)}\right)$, we want to prove that $p_c^{(g+1)} \sim \mathcal{N}\left(0, C^{(g)}\right)$:

$$
\begin{aligned}
p_c^{(g+1)} &= (1-c_c)p_c^{(g)} + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\frac{\sum_{i=1}^{\mu} w_i\left(x_i^{(g+1)} - m^{(g)}\right)}{\sigma^{(g)}} \\
&\sim (1-c_c)\mathcal{N}\left(0, C^{(g)}\right) + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\left(\sum_{i=1}^{\mu} w_i \mathcal{N}\left(0, C^{(g)}\right)\right) \\
&\sim \mathcal{N}\left(0, (1-c_c)^2 C^{(g)}\right) + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\mathcal{N}\left(0, \sum_{i=1}^{\mu} w_i^2 C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, (1-c_c)^2 C^{(g)}\right) + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\mathcal{N}\left(0, \frac{1}{\mu_{\text{eff}}} C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, (1-c_c)^2 C^{(g)}\right) + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\frac{1}{\sqrt{\mu_{\text{eff}}}}\mathcal{N}\left(0, C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, (1-c_c)^2 C^{(g)}\right) + \mathcal{N}\left(0, c_c(2-c_c) C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, \left((1-c_c)^2 + c_c(2-c_c)\right) C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, (1 - 2c_c + c_c^2 + 2c_c - c_c^2) C^{(g)}\right) \\
&\sim \mathcal{N}\left(0, C^{(g)}\right).
\end{aligned}
\tag{27}
$$

■