From Dependent Ciphertexts to Sharp Bounds for Polynomial Evaluation in BGV

Chiara Marcolla[1], Nadir Murru[2], and Matilda Urani[3]

[1] Technology Innovation Institute, Abu Dhabi, United Arab Emirates
[2] Università degli studi di Trento, Trento, Italy
[3] Politecnico di Torino, Torino, Italy

Abstract. A central challenge in Fully Homomorphic Encryption (FHE) is the selection of ciphertext moduli, which depends on accurately estimating noise growth during homomorphic evaluation. Choosing excessively large moduli reduces the efficiency and scalability of the scheme, whereas underestimating noise can result in decryption failures and potential security vulnerabilities. Average-case analyses have provided promising probabilistic estimates of noise growth in practical FHE computations. However, recent studies indicate that reliable estimates must account for dependencies among ciphertext errors, as neglecting these correlations can systematically underestimate noise.
This work presents the first average-case analysis for the BGV scheme that accounts for both correlations induced by secret/public keys and ciphertext dependencies. We propose new variance estimates for the noise coefficients in homomorphic additions and multiplications involving dependent ciphertexts, and extend our framework to the homomorphic evaluation of polynomial functions, which is widely employed in FHE applications. Our theoretical estimates closely align with experimental observations, exhibiting discrepancies smaller than 1.2 bits.
Finally, these estimates are adopted to choose the initial BGV parameters, leading to significantly smaller ciphertext moduli compared to those currently adopted in widely used FHE libraries, thus improving the efficiency of the scheme. Compared to OpenFHE, we obtain an average reduction of about 11.5% in the ciphertext modulus size, while relative to HElib, we reduce the ratio between consecutive moduli up to 23 bits per level in practical settings.

Keywords: Average-case · BGV · Fully Homomorphic Encryption · OpenFHE · Parameters Selection

1 Introduction

The increasing use of outsourced computation in settings such as cloud computing and machine learning raises fundamental concerns about data confidentiality. Fully Homomorphic Encryption (FHE) addresses precisely this issue by enabling computations directly over encrypted data. Since the first construction by Gentry in 2009 [22], an extensive body of research has led to several practical FHE schemes, including BGV [6], BFV [5,19], TFHE [18,11,12], and CKKS [10,9].

Current FHE constructions base their security on the assumed hardness of the Learning with Errors (LWE) problem [34], and its ring variant (RLWE) [28]. Informally, the RLWE problem consists of distinguishing samples of the form $(a, b = s \cdot a + e)$ from uniformly random samples over the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$, where correctness in FHE requires the error term e to remain sufficiently small compared to the ciphertext modulus q.

A fundamental challenge in FHE is therefore the selection of the ciphertext moduli. Choosing unnecessarily large moduli significantly increases the computational cost and memory consumption of the scheme, reducing its practicality and scalability. At the same time, moduli that are too small may compromise correctness or even security. Since the noise grows during homomorphic evaluation, parameter selection critically depends on obtaining accurate estimates of the error growth throughout the computation [29].

This issue is central to research in FHE and, over the years, various approaches have been proposed to address it. For example, employing the Euclidean norm [6], the infinity norm [19,26], and the canonical norm in *worst-case* analysis [13,15,17,23,25,30].

A promising approach in the recent literature is the adoption of the *average-case analysis*, which models the noise coefficients as random variables and derives bounds according to the characteristics of their distributions. Interest in this method, initially applied in the TFHE scheme [11], and subsequently in the CKKS [14,4], BGV [31,16,3] and BFV [2] schemes, grew due to a recognized discrepancy between worst-case analyses and experimental behavior, as highlighted in [15]. The introduction of the average-case approach offers a potential resolution to these disparities, as it makes possible to compute tight *probabilistic* upper bounds.

However, obtaining tight estimates is not sufficient. For parameter selection, it is equally important to guarantee that the noise is never underestimated, even marginally. Indeed, such underestimations can lead to two critical issues: first, ciphertexts may fail to decrypt correctly with non-negligible probability due to excessive noise; second, the scheme may become vulnerable to security attacks, as demonstrated in [7,8].

Recent works on BGV [3] and BFV [2] have shown that, in order to avoid such underestimations of the noise, it is crucial to account for the dependencies among the errors associated with the different ciphertexts involved in a computation. These dependencies arise because ciphertexts share common random components induced by the secret and public keys, making the corresponding noise terms inherently correlated. These dependencies are intrinsic to any circuit evaluated on encrypted data, as all ciphertexts are generated under the same key.

However, this is not the only source of dependence among the error terms. In fact, in practical computations, additional forms of dependence naturally arise. To understand this, it is necessary to distinguish between *common dependencies* and *ciphertext dependencies* as discussed in [21]. Common dependencies arise from the public and secret keys. Ciphertext dependencies, on the other hand, occur when the randomness involved in the encryption process is correlated,

for instance, in operations involving the same ciphertext, such as self-products. While for BFV [2] and CKKS [4], the study of specific circuits involving dependent ciphertexts has already been considered, only common dependencies are analyzed in the context of BGV. In fact, the authors of [3] focus only on the setting where the ciphertexts involved in the computation are mutually independent. However, in practical applications, homomorphic operations are frequently performed on dependent ciphertexts. Accurately accounting for such dependencies is therefore *essential* to correctly estimate the noise growth and to avoid systematic underestimations of the resulting error.

Understanding these phenomena is becoming increasingly important in modern FHE applications, where homomorphic computations often involve the evaluation of high-degree polynomial functions and deep arithmetic circuits. This setting repeatedly combines dependent ciphertexts, naturally introducing correlations among the resulting error terms. As the multiplicative depth grows, worst-case analyses tend to produce increasingly conservative noise estimates, leading to unnecessarily large ciphertext moduli and reduced efficiency. Therefore, accurate methods to estimate the error growth are fundamental to selecting ciphertext moduli more efficiently.

Our contributions. In this work, we present the first BGV average-case analysis that accounts for both *common dependencies*, induced by secret and public keys, and *ciphertext dependencies*, arising from the use of correlated ciphertexts in homomorphic computations.

First, we provide new estimates for the variance of the critical quantity after homomorphic additions (Proposition 3), multiplications (Theorems 2 and 3) involving potentially dependent ciphertexts, as well as after the product of a ciphertext and one of its rotations (Theorem 4). Our analysis provides bounds that closely match experimental observations, with discrepancies in the $\log_2$ of the variance consistently below 1.2 bits.

Second, we apply these techniques to the homomorphic evaluation of polynomials, where the same ciphertext is used many times and dependencies are therefore unavoidable. We derive noise estimates for two standard evaluation strategies, square-and-multiply (Theorem 5) and baby-step giant-step (Theorem 6). More generally, our framework applies to arbitrary polynomial evaluation strategies, once the corresponding sequence of homomorphic operations is specified. Indeed, the results developed in the previous theorems provide the basic building blocks needed to track the variance through additions, multiplications, and rotations of potentially dependent ciphertexts. This allow us not only to estimate the resulting noise growth, but also to compare alternative evaluation methods and identify the most efficient one for a given polynomial.

Finally, we leverage these estimates to improve BGV ciphertext modulus selection. Compared to the parameter choices adopted in current FHE libraries, our methodology achieves smaller ciphertext moduli while preserving correctness guarantees. In particular, when compared with OpenFHE [1], our methodology reduces the ciphertext modulus by an average of approximately 16.3 bits across all the evaluated benchmarks, corresponding to an average reduction of about

11.5%. Moreover, compared with the modulus ratios currently employed in HElib [36], our approach reduces the size of the ratios between consecutive moduli by up to 23 bits per level in practical parameter settings.

Roadmap of this work. The paper is structured as follows. Section 2 introduces the notation and the probabilistic tools used throughout the work. Section 3 gives an overview of the functioning of the BGV scheme. In Section 4 we provide accurate variance estimates of the critical quantity after the main homomorphic operations. Building upon these results, in Section 5, we explain how to correctly estimate the critical quantity resulting from the evaluation of generic polynomials. In Section 6 we compare our estimates with experimental results and we show how the results of previous sections can be leveraged to correctly select the ciphertext moduli. We also compare our parameters with those currently adopted in widely used FHE libraries. Finally, Section 7 concludes this work by summarizing the main results.

2 Preliminaries

2.1 Notation

Let n, t, and q be positive integers with n a power of two. We define $\mathcal{R}$ as the ring $\mathcal{R} = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $\mathcal{R}_d = \mathbb{Z}_d[x]/\langle x^n + 1\rangle$. We denote by M the multiplicative depth of the homomorphic circuit and by $L = M + 1$ the number of levels. We denote by t the plaintext modulus, which must satisfy $t \equiv 1 \mod 2n$, from which the plaintext space $\mathcal{R}_t$ is constructed. To support leveled homomorphic evaluation, we consider a chain of ciphertext moduli defined as

$$q_\ell = \prod_{j=0}^{\ell} p_j, \quad \ell \in \{0, \dots, L-1\},$$

where q_ℓ is the modulus used at level $L - 1 - \ell$. In particular, the initial modulus q_{L-1} (i.e., at level 0) will often be simply denoted by q. In the following, polynomials are denoted by lowercase letters, while vectors of polynomials are represented in bold. For a polynomial a, we denote by $a|_i$ the coefficient of x^i. We use $[\cdot]_q$ to indicate the centered reduction modulo q over the interval $[-q/2, q/2)$. When applied to a polynomial, the operation is intended to be coefficient-wise.

2.2 Probabilistic distributions

Given a polynomial $a \in \mathcal{R}$ and a probability distribution $\mathcal{X}$, the notation $a \leftarrow \mathcal{X}$ denotes that each coefficient of a is sampled independently according to $\mathcal{X}$. We denote by (i) $\mathcal{U}_q$ the uniform distribution over $\mathbb{Z}_q$ with representatives in $[-q/2, q/2)$; (ii) $\mathcal{DG}_q(0, \sigma^2)$ the discrete Gaussian distribution, obtained by sampling from a Gaussian distribution with mean 0 and variance σ^2, rounding to

the nearest integer, and reducing modulo q, with representatives in $[-q/2, q/2)$. Moreover, we denote by V_a the variance of the coefficients of a polynomial a and by $V(a|_i)$ the variance of the i–th coefficient. In particular, abusing the notation, we have that $V_{\mathcal{DG}_q(0,\sigma^2)} = \sigma^2$, $V_{\mathcal{U}_3} = 2/3$, and $V_{\mathcal{U}_q} = (q^2-1)/12 \approx q^2/12$. Finally, we recall the following well-known results.

Proposition 1 ([20]). *Let X, Y, Z, and W be random variables. Then,*

1. *if X, Y are independent w.r.t. Z, W, $\mathsf{Cov}(X, Y) = 0$ and $\mathbb{E}[X] = 0$, then $\mathsf{Cov}(XZ, YW) = 0$.*
2. *if X, Y are independent and $\mathbb{E}[X] = \mathbb{E}[Y] = 0$, then $\mathsf{Var}(XY) = \mathsf{Var}(X)\mathsf{Var}(Y)$.*
3. *if X, Y are independent then $\mathsf{Var}(X + Y) = \mathsf{Var}(X) + \mathsf{Var}(Y)$ otherwise $\mathsf{Var}(X + Y) = \mathsf{Var}(X) + \mathsf{Var}(Y) + 2\mathsf{Cov}(X, Y)$.*
4. *for any $\gamma \in \mathbb{Z}$, $\mathsf{Var}(\gamma X) = \gamma^2 \mathsf{Var}(X)$.*
5. *if X, Y have finite second moments, then $|\mathbb{E}[XY]| \leq \sqrt{\mathbb{E}[X^2] \cdot \mathbb{E}[Y^2]}$ (Cauchy–Schwarz Inequality).*

Lemma 1 ([37]). *Let n be a power of two and let $\zeta_1, \ldots, \zeta_n$ be the n roots of unity. Then, for every $z = \sum_{j=0}^{n-1} z|_j \, x^j \in \mathcal{R}$,*

$$\sum_{j=0}^{n-1} (z|_j)^2 \;=\; \frac{1}{n} \sum_{m=1}^{n} |z(\zeta_m)|^2 .$$

In particular, if z is a random polynomial whose coefficients have finite second moments,

$$\sum_{j=0}^{n-1} \mathbb{E}\left[(z|_j)^2\right] \;=\; \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}\left[|z(\zeta_m)|^2\right] .$$

3 The BGV scheme

In this section, we recall the three basic functions of the BGV scheme and the homomorphic operations of interest: addition, multiplication, modulus switching and rotation.

Key Generation.

Sample $s \leftarrow \chi_s$, $a \leftarrow \mathcal{U}_{q_{L-1}}$ and $e \leftarrow \chi_e$.
The key generation outputs the secret key $\mathsf{sk} = s$ and the public key $\mathsf{pk} = (b, a) = (-a \cdot s + te, a) \mod q_{L-1}$.

Encryption.

Given the plaintext $m \in \mathcal{R}_t$ and the public key $\mathsf{pk} = (b, a)$, sample $u \leftarrow \chi_s$ and $e_0, e_1 \leftarrow \chi_e$. The encryption outputs $\mathfrak{c} = (\boldsymbol{c}, q_{L-1}, \nu_{\mathsf{fresh}})$, where

$$\boldsymbol{c} = (c_0, c_1) = (bu + te_0 + m,\ au + te_1) \bmod q_{L-1},$$

where $u, e_0, e_1 \in \mathcal{R}_{q_{L-1}}$ and $\boldsymbol{c} \in \mathcal{R}^2_{q_{L-1}}$.

Decryption.

Given $\mathfrak{c} = (\boldsymbol{c}, q_\ell, \nu)$ and the secret key $\mathsf{sk} = s$.
The decryption outputs $m = \Big[[c_0 + c_1 \cdot s]_{q_\ell}\Big]_t$.

We adopt the term *extended ciphertext* to refer to the tuple $\mathfrak{c} = (\boldsymbol{c}, q_\ell, \nu)$ where $\boldsymbol{c}$ is the ciphertext, q_ℓ is the corresponding modulus, and ν is the associated critical quantity. The *critical quantity* $\nu = [c_0 + c_1 \cdot s]_{q_\ell}$ of a ciphertext $\boldsymbol{c}$ defines whether $\boldsymbol{c}$ can be correctly decrypted. A proof of the correctness of the decryption process can be easily verified for a fresh ciphertext. In fact, the critical quantity can be rewritten as

$$\nu_{\mathsf{fresh}} = [c_0 + c_1 \cdot s]_{q_{L-1}} = [m + t(e \cdot u + e_1 \cdot s + e_0)]_{q_{L-1}} = [m + t\varepsilon]_{q_{L-1}},$$

where ε denotes the *error* introduced during encryption. Since the critical quantity is simply the message perturbed by an error term whose coefficients are multiples of t, it is immediate to verify that reduction modulo t recovers the original message if the error remains below a certain threshold. Precisely, to ensure the correctness of the decryption, it is necessary that [30]: $\|\nu\|_\infty < \frac{q_\ell}{2}$. Otherwise, $m + t\varepsilon$ may wrap around the modulus, causing decryption to output a random polynomial rather than the original message.

When operations are performed on ciphertexts, they are reflected in the associated critical quantity, which grows with each operation [13]. Therefore, to correctly select the ciphertext moduli, it is fundamental to accurately estimate the magnitude of this critical quantity, commonly referred to as the *noise*.

Variance of the error coefficients. Adopting an average-case perspective, we are thus interested in estimating the variance of the coefficients of the critical quantity ν. Indeed, this variance allows us to derive tight probabilistic bounds on the magnitude of the noise and, consequently, to select ciphertext moduli that are as small as possible while still ensuring correct decryption.
For the critical quantity of a fresh ciphertext, we have that [3]

$$V_{\mathsf{clean}} = \mathsf{Var}(\nu_{\mathsf{fresh}}|_i) = t^2 \left(\tfrac{1}{12} + nV_eV_u + V_e + nV_eV_s\right). \tag{1}$$

Homomorphic addition.

Given two extended ciphertexts $(\boldsymbol{c}, q_\ell, \nu)$ and $(\boldsymbol{c}', q_\ell, \nu')$, the homomorphic addition outputs $(\boldsymbol{c}_{\mathsf{add}}, q_\ell, \nu_{\mathsf{add}})$, where

$$\boldsymbol{c}_{\mathsf{add}} = (c_0 + c_0', c_1 + c_1') \mod q_\ell,$$

with $\nu_{\mathsf{add}} = \nu + \nu'$.

We are going to study the variance $\mathsf{Var}(\nu_{\mathsf{add}}|_i)$ in Propositions 2 and 3.

Homomorphic multiplication.

Given two extended ciphertexts $(\boldsymbol{c}, q_\ell, \nu)$ and $(\boldsymbol{c}', q_\ell, \nu')$, the homomorphic multiplication outputs $((d_0, d_1, d_2), q_\ell, \nu_{\mathsf{mul}})$, where

$$(d_0, d_1, d_2) = (c_0 \cdot c_0', c_0 \cdot c_1' + c_1 \cdot c_0', c_1 \cdot c_1') \mod q_\ell,$$

with $\nu_{\mathsf{mul}} = \nu \cdot \nu' = d_0 + d_1 s + d_2 s^2$.

It is worth noting that the product of two ciphertexts (each represented by two polynomials) outputs a ciphertext represented by three polynomials, thereby violating the compactness property. This increases the cost of subsequent operations, making it necessary to transform the ciphertext (d_0, d_1, d_2) back into the form $(\bar{c}_0, \bar{c}_1)$. To achieve this, a technique known as *relinearization*, or *key switching*, is employed.

It is worth noting that in [3, Theorem 2], the authors proposed a way to correctly estimate the coefficients variance of $\nu \cdot \nu'$, which will be further developed and employed later in this work (see Theorems 2 and 3).

Modulus switching.

Given an extended ciphertext $(\boldsymbol{c}, q_\ell, \nu)$, and given an integer $q_{\ell'}$ such that $q_{\ell'} < q_\ell$, the modulus switching procedure outputs $(\boldsymbol{c}', q_{\ell'}, \nu_{\mathsf{ms}})$, where

$$\boldsymbol{c}' = \frac{q_{\ell'}}{q_\ell}(\boldsymbol{c} + \boldsymbol{\delta}) \mod q_{\ell'}, \text{ with } \boldsymbol{\delta} = t[-\boldsymbol{c}t^{-1}]_{q_\ell/q_{\ell'}}.$$

This modulus switching technique allows us to reduce the size of the noise associated with a given ciphertext. Indeed, the output ciphertext still decrypts to the original plaintext (scaled by a factor of $q_\ell/q_{\ell'}$), but has a smaller associated noise. For this reason, it is typically applied to ciphertexts involved in homomorphic multiplications, where noise growth is most significant.

The critical quantity associated with the new ciphertext $\boldsymbol{c}'$ can be expressed in terms of that of $\boldsymbol{c}$, namely,

$$\nu_{\mathsf{ms}} = [c_0' + c_1' \cdot s]_{q_{\ell'}} = \frac{q_{\ell'}}{q_\ell}([c_0 + c_1 \cdot s]_{q_\ell} + \delta_0 + \delta_1 \cdot s) = \frac{q_{\ell'}}{q_\ell}(\nu + \delta_0 + \delta_1 \cdot s).$$

As shown in [3] the term δ_0 is negligible and the coefficients variance of ν_{ms} can be estimated as

$$\begin{aligned}\mathsf{Var}(\nu_{\mathsf{ms}}|_i) &= \left(\frac{q_{\ell'}}{q_\ell}\right)^2 (\mathsf{Var}(\nu|_i) + \mathsf{Var}(\delta_1 \cdot s|_i)) \\ &= \left(\frac{q_{\ell'}}{q_\ell}\right)^2 (\mathsf{Var}(\nu|_i) + nV_{\delta_1}V_s)\,. \end{aligned} \tag{2}$$

Rotations

Let $\sigma_k : \mathcal{R}_q \to \mathcal{R}_q$ be the ring automorphism defined by $\sigma_k(x) : x \to x^k$, where k is an odd integer. Given an extended ciphertext $(\boldsymbol{c}, q_\ell, \nu)$ the rotation operation consists in applying the automorphism to both the ciphertext components, i.e.,

$$\sigma_k(\boldsymbol{c}) = (\sigma_k(c_0), \sigma_k(c_1)) \mod q_\ell.$$

The critical quantity associated with the rotated ciphertext is given by

$$\sigma_k(\nu) = \sigma_k(c_0) + \sigma_k(c_1)\sigma_k(s).$$

Since $\sigma_k(\cdot)$ permutes the coefficients up to signs, the coefficient variance is unchanged, so

$$\mathsf{Var}(\sigma_k(\nu)|_i) = \mathsf{Var}(\nu|_j),$$

for a coefficient j established by the automorphism. Note that the resulting ciphertext encrypts the rotated plaintext with respect to $\sigma_k(s)$. To obtain a ciphertext encrypting the same plaintext but with respect to the initial secret key s, the key switching technique is employed. Therefore, throughout this work we will always assume that a key switching operation is performed after rotation. This is consistent with the behavior typically observed in the most common FHE libraries. For instance, in OpenFHE, when the operation `EvalRotate` is used, an implicit key-switching operation is also performed.

Intuitively, the basic idea of key switching is to use an encryption of a new secret key sk' under the original secret key sk to transform a ciphertext associated with sk' into an equivalent ciphertext encrypted under sk.

It is commonly applied in two scenarios. The first occurs after the multiplication of two ciphertexts: the resulting ciphertext has three components (c_0, c_1, c_2), and its decryption involves the secret key powers 1, s, and s^2. In this case, $\mathsf{sk}' = s^2$, and key switching transforms it into an equivalent ciphertext with only two components, $\bar{\boldsymbol{c}} = (\bar{c}_0, \bar{c}_1)$, encrypted under the original secret key $\mathsf{sk} = s$.

The second occurs after a rotation, which transforms the ciphertext components into $(\sigma_k(c_0), \sigma_k(c_1))$, resulting in a ciphertext associated with the rotated secret key $\mathsf{sk}' = \sigma_k(s)$. In this case, key switching transforms the term $\sigma_k(c_1)\sigma_k(s)$ into an equivalent representation under the original secret key, where, in this case, $c_2 = \sigma_k(c_1)$.

Key Switching (Intuitive Idea)

Let $(\boldsymbol{c}, q_\ell, \nu)$ be a ciphertext where $\boldsymbol{c} = (c_0, c_1, c_2)$, and c_2 is associated with a secret key sk'. The key switching operation generates a key switching key

$$\mathsf{ksk} = (\mathsf{ksk}_0, \mathsf{ksk}_1) = (-a \cdot \mathsf{sk} + te + \mathsf{sk}', a) \mod q_\ell,$$

where $a \leftarrow \mathcal{U}_{q_\ell}$ and $e \leftarrow \chi_e$, and outputs $(\bar{\boldsymbol{c}}, q_\ell, \nu_{\mathsf{ks}})$, with

$$\bar{\boldsymbol{c}} = (c_0 + c_2 \cdot \mathsf{ksk}_0, c_1 + c_2 \cdot \mathsf{ksk}_1) \mod q_\ell.$$

Note that, this specific construction is only intended as an informal way of illustrating key switching and is not used in practice, as its error contribution would be excessive. However, in this work, we do not need to specify the details of the method, nor we need to explicitly track the associated error contribution. Indeed, when applied after multiplication, its error contribution can be neglected, see [3, Appendix C]. For rotations, the key switching operation we consider is always followed by a modulus switching step, which cancels the contribution of the key switching error, as explained in Section 4. Hence, throughout this work, we abstract away the specific implementation of key switching.

4 Average-Case Noise Analysis

In this section, we present our results on how to precisely estimate the variance of the error coefficients arising from the addition and product of *dependent* ciphertexts. Most existing average-case analyses rely on the strong assumption that the error terms involved in a computation are independent. However, as shown in [3] this does not hold in the BGV context. Indeed, the authors show that even when ciphertexts are independently generated, the associated critical quantities are not independent, since they share structural components induced by the public and secret keys. We refer to these dependencies as *common dependencies*, according to the terminology introduced in [21]. Therefore, the authors of [3] provided tight estimation bounds able to capture the effect of these common terms. However, beyond these, one must also consider *ciphertext dependencies* [21] which arise when the ciphertexts involved in a computation are themselves dependent. This situation naturally occurs in real-world computations, where operations are often performed on dependent data.

However, the framework of [3] only discusses common dependencies, and explicitly states that this second source of correlation is not addressed. Accounting for both types of dependencies is essential, as they increase the variance of the coefficients of the critical quantities involved. Ignoring them leads to an underestimation of the error, which must be avoided not only to ensure the correct functioning of the scheme but also for security reasons, as shown in [7,8]. In this section, we fill this gap by providing a precise analysis of the variance of the coefficients of the critical quantity able to account both for common and ciphertext dependencies.

Furthermore, we explicitly examine how rotations of a ciphertext influence the effects of these dependencies. As discussed in Section 5, accurately estimating this effect is important not just because rotations can occur within arbitrary computational circuits but also since several recent methods designed to optimize polynomial evaluation directly utilize this operation. Consequently, a precise characterization is essential to the scope of this work.

This section is organized as follows. In Section 4.1 we recall the main results needed to study common dependencies in the case of independent ciphertexts, which will also be used in the analysis of ciphertext dependencies. In Section 4.2, we derive variance estimates for the critical quantity arising from products and additions of dependent ciphertexts. Finally, in Section 4.3, we accurately estimate the variance resulting from the product of a ciphertext by a rotation of itself.

4.1 Tracking Common Dependencies

We briefly recall the main results of [3], which provide the baseline for our analysis.

Notation 1 *Each critical quantity involved in a circuit of homomorphic operations can be expressed as a polynomial in the secret and public keys, as follows*

$$\nu = \sum_{\iota} a_{\iota} s^{\iota} = \sum_{\iota} \sum_{\mu} b_{\mu}(\iota) e^{\mu} s^{\iota},$$

where $a_{\iota} = \sum_{\mu} b_{\mu}(\iota) e^{\mu}$*, and* $b_{\mu}(\iota)$ *contains no powers of* s *or* e*.*

Lemma 2 ([3, Lemma 3]). *Let* $a(x) = a|_0 + a|_1 x + \cdots + a|_{n-1} x^{n-1} \in \mathcal{R}$*, where* $a|_i$*'s are i.i.d. random variables with* $\mathbb{E}[a|_i] = 0$*,* $\mathbb{E}[a|_i^2] = V_a$*. Then, as* n *approaches infinity, for all* $0 \leq i \leq n-1$ *we have*

$$\mathbb{E}[a^k|_i] = 0, \quad \mathbb{E}[(a^k|_i)^2] = k! n^{k-1} V_a^k \,,$$

where $a^k|_i$ *denotes the* i*–th coefficient of the polynomial* $a(x)^k$*.*

Remark 1. Lemma 2 is an asymptotic statement. However, since BGV typically uses ring dimension $n \geq 2^{13}$, the error introduced by the asymptotic approximation is negligible and the resulting expressions, which we use throughout the paper, can be considered reliable.

For clarity of exposition, the main results proved in [3, Section 6] are summarized in the following statements.

Lemma 3. *Let* $(\boldsymbol{c}, q_\ell, \nu)$ *be an extended ciphertext and let* $\nu_{\mathsf{ms}} = \frac{q'_\ell}{q}\big(\nu + \delta_1 s\big)$*, be the critical quantity after modulus switching to the target modulus* q'_ℓ*, where* $q = pq'_\ell$*. If* p *is chosen sufficiently large so that* $\mathsf{Var}(\nu/p|_i) \ll \mathsf{Var}(\delta_1 s/p|_i)$*, then the critical quantity can be approximated as*

$$\nu_{\mathsf{ms}} \approx \delta_1 s/p, \tag{3}$$

where $\delta_1/p \leftarrow \mathcal{U}_t$. *In particular, the coefficients of the critical quantity after modulus switching have variance*

$$V_{\mathsf{ms}} := nV_{\delta_1/p}V_s = n\frac{t^2}{12}V_s\,. \tag{4}$$

Thus, the variance after modulus switch no longer depends on the initial ciphertext considered.

In this work, we always assume that whenever modulus switching is applied, the ratio between the moduli is sufficiently large, as required by the condition above. This is a realistic assumption for the parameter choices used in practical applications today, see [3] for an in-depth discussion.

Theorem 1. *The product of two independent ciphertexts that have undergone modulus switching has coefficient-wise variance* $2nV_{\mathsf{ms}}^2$.

Intuitively, this factor 2 corresponds exactly to the correction discussed at the beginning of this section, which accounts for the dependencies introduced by the keys used in the scheme. In particular, we are considering here the product of two ciphertexts whose associated critical quantities are of the form $\delta_1 s/p$ and $\delta_1' s/p$, where δ_1 and δ_1' are mutually independent. The factor 2 precisely captures the increase in variance caused by the fact that both expressions share the same secret key s. If the two ciphertexts were totally independent, this additional contribution would not appear. More generally, the correction factor grows with the number of terms shared by the critical quantities involved in the multiplication, as shown in the following section.

Proposition 2 ([3, Proposition 2]). *Let* $\boldsymbol{c}$ *and* $\boldsymbol{c}'$ *be two independent ciphertexts with critical quantities* ν, *and* ν', *respectively. Then, the variance of the error coefficients resulting from the addition of* $\boldsymbol{c}$ *and* $\boldsymbol{c}'$ *is*

$$\mathsf{Var}(\nu_{\mathsf{add}}|_i) = \mathsf{Var}(\nu|_i) + \mathsf{Var}(\nu'|_i). \tag{5}$$

4.2 Tracking Ciphertext Dependencies

In this section, we consider the worst-case scenario in which the number of dependencies is maximized, namely the squaring of a ciphertext.

Theorem 2. *Let* $(\boldsymbol{c}, q_\ell, \nu_{\mathsf{ms}})$ *be an extended ciphertext, that has just undergone modulus switching. Let* $(\boldsymbol{c^{(2)}}, q_\ell, \nu_{\mathsf{mul}})$ *be the result of squaring the ciphertext* $\boldsymbol{c}$. *Then, the variance of the coefficients of the resulting critical quantity* ν_{mul} *can be estimated as*

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) \approx 4nV_{\mathsf{ms}}^2 \tag{6}$$

Intuition. In contrast to the independent case, squaring introduces additional dependencies, since both operands share, not only the secret key s as before, but also the same term $\frac{\delta_1}{p}$, which is uniformly distributed. This results in an additional factor 2 in the variance (see Theorem 1).

Proof. By Lemma 3, we can write $\nu_{\mathsf{ms}} \approx \frac{\delta_1}{p} \cdot s$ and

$$\mathsf{Var}(\nu_{\mathsf{ms}}|_i) = V_{\mathsf{ms}} = nV_{\frac{\delta_1}{p}} V_s.$$

Our aim is to precisely define the variance of the coefficients of the critical quantity ν_{mul} associated to $\boldsymbol{c^{(2)}}$, i.e. of $\nu^2_{\mathsf{ms}}|_i$. We start by writing $\nu^2_{\mathsf{ms}}|_i$ as

$$\nu^2_{\mathsf{ms}}|_i = \left(\frac{\delta_1}{p} s\right)^2 \Big|_i = \sum_{j=0}^{n-1} \xi(i,j) \left(\frac{\delta_1}{p}\right)^2 \Big|_j \cdot s^2|_{i-j}$$

We want to estimate

$$\mathsf{Var}\left(\left(\frac{\delta_1}{p} s\right)^2 \Big|_i\right) = \mathsf{Var}\left(\sum_{j=0}^{n-1} \left(\frac{\delta_1}{p}\right)^2 \Big|_j \cdot s^2|_{i-j}\right)$$

For the sake of clarity, let $X_i = (\frac{\delta_1}{p})^2|_i$ and $Y_i = s^2|_i$. We know, according to Lemma 2 that $\mathbb{E}[X_i] = \mathbb{E}[Y_i] = 0\,, \mathsf{Var}(X_i) = 2nV^2_{\frac{\delta_1}{p}}$, and $\mathsf{Var}(Y_i) = 2nV_s^2$. We want to study

$$\begin{aligned}\mathsf{Var}(\sum_j X_j Y_{i-j}) = \sum_j \mathsf{Var}(X_j)\mathsf{Var}(Y_{i-j}) + \\ + \sum_{j_1 \neq j_2} \xi(i,j_1)\xi(i,j_2)\mathsf{Cov}(X_{j_1}Y_{i-j_1}, X_{j_2}Y_{i-j_2}). \qquad (7)\end{aligned}$$

It can be proven that $\mathsf{Cov}(X_{j_1}Y_{i-j_1}, X_{j_2}Y_{i-j_2}) = 0$, for $j_1 \neq j_2$. In fact, it is immediate to verify that $X_{j_1}, X_{j_2} \perp Y_{i-j_1}Y_{i-j_2}$, and $\mathbb{E}[X_{j_1}] = 0$. Moreover, it is possible to show that $\mathsf{Cov}(X_{j_1}, X_{j_2}) = 0$ for $j_1 \neq j_2$, thereby concluding $\mathsf{Cov}(X_{j_1}Y_{i-j_1}, X_{j_2}Y_{i-j_2}) = 0$, according to Proposition 1 (1). Indeed,

$$\begin{aligned}\mathsf{Cov}(X_{j_1}, X_{j_2}) = \mathbb{E}[X_{j_1}X_{j_2}] = \mathbb{E}\left[\left(\frac{\delta_1}{p}\right)^2 \Big|_{j_1} \left(\frac{\delta_1}{p}\right)^2 \Big|_{j_2}\right] = \\ = \frac{1}{p^4}\mathbb{E}[\sum_{k_1} \xi(j_1,k_1)\delta_1|_{k_1}\delta_1|_{j_1-k_1} \sum_{k_2} \xi(j_2,k_2)\delta_1|_{k_2}\delta_1|_{j_2-k_2}] = \\ = \frac{1}{p^4}\sum_{k_1}\sum_{k_2} \xi(j_1,k_1)\xi(j_2,k_2)\mathbb{E}[\delta_1|_{k_1}\delta_1|_{j_1-k_1}\delta_1|_{k_2}\delta_1|_{j_2-k_2}].\end{aligned}$$

We are considering $j_1 \neq j_2$, hence we have two possible cases: $k_1 = k_2$ or $k_1 \neq k_2$.

- When $k_1 = k_2 = k$ then $j_1 - k \neq j_2 - k$, hence at least one among $\delta_1|_{j_1-k}, \delta_1|_{j_2-k}$ is different (and so independent) from $\delta_1|_k$, thus
$$\mathbb{E}[\delta_1|_k^2 \delta_1|_{j_1-k}\delta_1|_{j_2-k}] = \mathbb{E}[\delta_1|_{j_1-k}]\mathbb{E}[\delta_1|_k^2\delta_1|_{j_2-k}] = 0.$$
- When $k_1 \neq k_2$ then $\xi(j_1,k_1)\xi(j_2,k_2)\mathbb{E}[\delta_1|_{k_1}\delta_1|_{j_1-k_1}\delta_1|_{k_2}\delta_1|_{j_2-k_2}]$. Again, if one term is different from the others, the total value is zero. Therefore, the only way this can be different from zero is if the indexes are pairwise equal (modulo n). In such case we get $\mathbb{E}[\delta_1|_{k_1}^2\delta_1|_{k_2}^2]$. Again, we have two possibilities

- $k_1 = j_2 - k_2$ and $k_2 = j_1 - k_1$, which however means that $j_1 = j_2$ which is a contradiction and therefore cannot occur.
- $k_1 = j_1 - k_1$ and $k_2 = j_2 - k_2$, which means that $j_1 = 2k_1 \mod n$ and $j_2 = 2k_2 \mod n$. Since n is even, if there is a solution $k_i = j_i/2$ then also $k_i = j_i/2 + n/2$ is a solution. For one of them, we obtain a positive sign $\xi(j_1, k_1) = 1$, whereas for the other we obtain a negative sign $\xi(j_1, k_1) = -1$. As a consequence, the two contributions cancel each other out, yielding a total contribution of zero also in these cases.

Therefore, according to Proposition 1 (2), Equation (7) becomes

$$\begin{aligned}\mathsf{Var}(\sum_j X_j Y_{i-j}) &= \sum_j \mathsf{Var}(X_j Y_{i-j}) = \sum_j \mathsf{Var}(X_j)\mathsf{Var}(Y_{i-j}) \\ &= n(2nV^2_{\frac{\delta_1}{p}} \cdot 2nV_s^2) = 4n(nV_{\frac{\delta_1}{p}} \cdot V_s)^2 = 4nV^2_{\mathsf{ms}},\end{aligned}$$

where V_{ms} is as in Equation (4). □

When considering higher powers of the same ciphertext, one would naturally expect a significant increase in dependencies among the resulting terms. Indeed, repeated multiplications of the same ciphertext introduce a large number of shared monomials in the underlying polynomial representation of the critical quantity. These induce additional correlations which, in principle, should lead to a more intricate variance structure and potentially larger coefficient growth, as happens, for example, in the BFV scheme [2].

However, the effect of these dependencies is substantially simplified in real-world scenarios, thanks to the use of modulus switching. As shown in Lemma 3, after modulus switching the critical quantity admits the approximation

$$\nu_{\mathsf{ms}} \approx \delta_1 s/p.$$

where $\frac{\delta_1}{p}$ is uniformly distributed and, crucially, is no longer affected by dependencies caused by the secret/public keys or, most importantly for our analysis, by the randomness shared among dependent ciphertexts.

Theorem 3. *Let $(\boldsymbol{c}_1, q, \nu_1)$ and $(\boldsymbol{c}_2, q, \nu_2)$ be two (possibly dependent) extended ciphertexts that have just undergone modulus switching. Let $(\boldsymbol{c}_{\mathsf{mul}}, q, \nu_{\mathsf{mul}})$ denote the resulting ciphertext. Then,*

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) \leq 4nV^2_{\mathsf{ms}}.$$

Proof. Since both $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ have just undergone modulus switch, according to Lemma 3 the associated critical quantities are of the form

$$\nu_1 \approx \frac{1}{p_1}\delta_1{}^{(\boldsymbol{c_1})} \cdot s \quad \text{and} \quad \nu_2 \approx \frac{1}{p_2}\delta_1{}^{(\boldsymbol{c_2})} \cdot s.$$

Define $X_1 = \frac{\delta_1{}^{(\boldsymbol{c_1})}}{p_1}, X_2 = \frac{\delta_1{}^{(\boldsymbol{c_2})}}{p_2}, Z = X_1 \cdot X_2, S = s^2$, it holds that,

$$\nu_{\mathsf{mul}} = Z \cdot S,$$

and we are interested in bounding the following quantity

$$\mathsf{Var}(Z \cdot S|_i) = \mathsf{Var}(\sum_{j=0}^{n-1} \xi(i,j) Z|_j \cdot S|_{i-j}) = \sum_{j=0}^{n-1} \mathsf{Var}(Z|_j \cdot S|_{i-j}),$$

where the last equality holds since all the covariance terms vanish by Proposition 1 (1). Indeed, S is independent of Z, $\mathbb{E}[S|_{i-j_1}] = 0$ by Lemma 2, and $\mathsf{Cov}(S|_{i-j_1}, S|_{i-j_2}) = 0$, as shown in the proof of Theorem 2. Using again that S and Z are independent, and that $\mathbb{E}[S|_i^2] = 2nV_s^2, \mathbb{E}[S|_i] = 0$ by Lemma 2 we obtain

$$\mathsf{Var}(Z|_j \cdot S|_{i-j}) = \mathbb{E}[Z|_j^2] \cdot \mathbb{E}[S|_{i-j}^2] - (\mathbb{E}[Z|_i] \cdot \mathbb{E}[S|_{i-j}])^2 = 2nV_s^2 \cdot \mathbb{E}[Z|_j^2],$$

thus,

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) = \mathsf{Var}(Z \cdot S|_i) = \sum_{j=0}^{n-1} 2nV_s^2 \cdot \mathbb{E}[Z|_j^2] = 2nV_s^2 \sum_{j=0}^{n-1} \mathbb{E}[Z|_j^2].$$

It remains to show that $\sum_{j=0}^{n-1} \mathbb{E}[Z|_j^2] \leq n\left(2nV_{X_1}^2\right)$. In such case,

$$\mathsf{Var}(\nu_{\mathsf{mul}}|_i) \leq 4n(n^2 V_s^2 V_{X_1}^2) = 4nV_{\mathsf{ms}}^2.$$

Using Lemma 1, Proposition 1 (5) and that the canonical embedding is a homomorphism, it holds that,

$$\begin{aligned}
\sum_{j=0}^{n-1} \mathbb{E}[Z|_j^2] &= \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}[|Z(\zeta_m)|^2] = \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}[|X_1(\zeta_m)|^2 \cdot |X_2(\zeta_m)|^2] \\
&\leq \frac{1}{n} \sum_{m=1}^{n} \sqrt{\mathbb{E}[|X_1(\zeta_m)|^4] \cdot \mathbb{E}[|X_2(\zeta_m)|^4]} = \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}[|X_1(\zeta_m)|^4] \\
&= \frac{1}{n} \sum_{m=1}^{n} \mathbb{E}[|X_1^2(\zeta_m)|^2] = \sum_{j=0}^{n-1} \mathbb{E}[X_1^2|_j^2] = n(2nV_{X_1}^2),
\end{aligned}$$

thus concluding the proof.

With the purpose of evaluating any function of the form $f(x) = \sum_{j=1}^{\kappa} a_j x^j$, we also need to introduce the result below, regarding the addition of (possibly) dependent ciphertexts.

Proposition 3. *Let $(\boldsymbol{c}_1, q_1, \nu_1)$ and $(\boldsymbol{c}_2, q_2, \nu_2)$ be two (possibly dependent) extended ciphertexts. Then the variance of the error coefficients ν_{add} resulting from the addition of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ is*

$$\mathsf{Var}(\nu_{\mathsf{add}}|_i) \leq \mathsf{Var}(\nu_1|_i) + \mathsf{Var}(\nu_2|_i) + 2\sqrt{\mathsf{Var}(\nu_1|_i)\mathsf{Var}(\nu_2|_i)}$$

In particular, let κ_1, κ_2 be two distinct positive integers. If $\boldsymbol{c}_1 = \boldsymbol{c}^{\kappa_1}$ and $\boldsymbol{c}_2 = \boldsymbol{c}^{\kappa_2}$ are obtained by computing powers of a ciphertext $\boldsymbol{c}$, then

$$\mathsf{Var}(\nu_{\mathsf{add}}|_i) \approx \begin{cases} \mathsf{Var}(\nu_1|_i) + V_{\mathsf{ms}} & \text{if } q_1 < q_2 \\ \mathsf{Var}(\nu_1|_i) + \mathsf{Var}(\nu_2|_i) & \text{if } q_1 = q_2 \end{cases}.$$

Proof. From point (3) of Proposition 1, it holds that

$$\mathsf{Var}(\nu_{\mathsf{add}}|_i) = \mathsf{Var}(\nu_1|_i) + \mathsf{Var}(\nu_2|_i) + 2\mathsf{Cov}(\nu_1|_i, \nu_2|_i).$$

Since $\mathsf{Var}(\nu_1|_i)$ and $\mathsf{Var}(\nu_2|_i)$ are finite, then $\mathsf{Cov}(\nu_1|_i, \nu_2|_i) \leq \sqrt{\mathsf{Var}(\nu_1|_i)\mathsf{Var}(\nu_2|_i)}$. In particular, let $\boldsymbol{c}_1 = \boldsymbol{c}^{\kappa_1}$ and $\boldsymbol{c}_2 = \boldsymbol{c}^{\kappa_2}$ with $\kappa_1 \neq \kappa_2$.

- If $q_1 < q_2$, then $\boldsymbol{c}^{\kappa_2}$ is first subjected to a modulus switch to q_1. Hence, according to Lemma 3 the variance of the critical quantity can be approximated by V_{ms}. Note that we always assume to perform a modulus switch before multiplication, thus ν_1 is the product of terms as in Equation (3). By applying the same argument as in Theorem 2, it is straightforward to see that the covariance is null, which concludes the first part of the proof.
- If $q_1 = q_2$, then there is no need to apply modulus switching to q_1. Now, both ν_1, ν_2 are products of terms as in Equation (3) and again by applying the same reasoning it can be shown that the covariance vanishes, concluding the proof.

□

4.3 Rotations

Finally, we study how dependencies influence the variance resulting from the product of a ciphertext with one of its rotations. This scenario is of particular interest not just because it may appear in a generic homomorphic evaluation circuit, but especially because some recent methods that optimize polynomial evaluation [32,39] directly involve computations of this form, as will be discussed later. In particular, one may ask whether the product of a ciphertext and its rotation behaves more like the product of two independently generated ciphertexts or like the squaring of a single ciphertext. In this section, we show that modulus switching, combined with key switching, makes the product of a ciphertext and its rotation resemble the first scenario. To introduce this result, formalized in Theorem 4, we first introduce an auxiliary lemma.

Lemma 4. *Let p be a prime and let $J \subseteq \mathcal{R}_p$ be an ideal with $\dim_{\mathbb{F}_p} J > n/2$. Then, for every pair of distinct indices $j_1 \neq j_2$, the coordinate projection*

$$\pi : J \longrightarrow \mathbb{F}_p^2, \qquad a \longmapsto \big(a|_{j_1}, a|_{j_2}\big)$$

is surjective, i.e. $\pi(J) = \mathbb{F}_p^2$.

Proof. We refer the reader to Appendix A for the proof of the lemma.

Theorem 4. *Let k be an odd integer and let $\sigma_k : \mathcal{R}_{pq} \mapsto \mathcal{R}_{pq}$ be the ring automorphism defined by $\sigma_k : x \mapsto x^k$. Let $(\mathbf{c}, pq, \nu)$ be an extended ciphertext. Assume that $\sigma_k(\mathbf{c})$ first undergoes key switching to the original secret key, yielding $\bar{\mathbf{c}}$. Then $\mathbf{c}$ and $\bar{\mathbf{c}}$ undergo modulus switching and are multiplied. Denote by $(\mathbf{c}_{\sigma_k}, q, \nu_{\sigma_k})$ the result of these operations. Then*

$$\mathrm{Var}\big(\nu_{\sigma_k}|_i\big) \approx 2n\, V_{ms}^2.$$

Proof. We start by observing that before modulus switching, the rotated ciphertext must go through a key switching procedure, since $\sigma_k(\mathbf{c})$ is a ciphertext under the rotated secret key $\sigma_k(s)$. However, we do not need to explicitly compute the noise introduced by key switching. Indeed, by Lemma 3 we know that, after modulus switch, the dominant component of the error is given by the correction term introduced by modulus switching itself (i.e. the term of the form δs), which dominates the contribution of the previous noise, making it negligible. Thus, we are going to consider the very naive key switching technique, which simply consists in encrypting the old secret key. After key switching, the rotated ciphertext is given by

$$\bar{\boldsymbol{c}} = (\sigma_k(c_0) + \sigma_k(c_1) \cdot \mathsf{ksk}_0, \sigma_k(c_1) \cdot \mathsf{ksk}_1) \mod pq,$$

where

$$\mathsf{ksk} = (\mathsf{ksk}_0, \mathsf{ksk}_1) = (-a_{\mathsf{ks}} s + t e_{\mathsf{ks}} + \sigma_k(s), a_{\mathsf{ks}}),$$

is the key switching key and $a_{\mathsf{ks}} \leftarrow \mathcal{U}_{pq}$ and $e_{\mathsf{ks}} \leftarrow \chi_e$. Before multiplication, both $\boldsymbol{c}$ and $\bar{\boldsymbol{c}}$ are modulus switched to q. Denote by

$$\boldsymbol{\delta}^{(\boldsymbol{c})} := t[-\boldsymbol{c}t^{-1}]_p \quad \text{and} \quad \boldsymbol{\delta}^{(\bar{\boldsymbol{c}})} := t[-\bar{\boldsymbol{c}}t^{-1}]_p,$$

the correction terms introduced by modulus switching, where, by construction, the quantities $[-\boldsymbol{c}t^{-1}]_p$ and $[-\bar{\boldsymbol{c}}t^{-1}]_p$ are RLWE samples. By Lemma 3, after modulus switching to q, the critical quantities associated with the two ciphertexts are given by

$$\nu \approx \frac{1}{p} {\delta_1}^{(\boldsymbol{c})} \cdot s \quad \text{and} \quad \bar{\nu} \approx \frac{1}{p} {\delta_1}^{(\bar{\boldsymbol{c}})} \cdot s.$$

Denote by $X_i := \frac{1}{p^2} {\delta_1}^{(\bar{\boldsymbol{c}})} {\delta_1}^{(\boldsymbol{c})}|_i$, $Y_i := s^2|_i$. Following the same reasoning of Theorem 2, the variance of the coefficients of $\nu_{\sigma_k} = \nu \cdot \bar{\nu}$ can be written as

$$\mathsf{Var}(\nu_{\sigma_k}|_i) = \mathsf{Var}(\sum_j X_j Y_{i-j}) = \sum_j \mathsf{Var}(X_j)\mathsf{Var}(Y_{i-j}) = n(\mathsf{Var}(X_j) \cdot 2nV_s^2).$$

Hence, the proof consists in showing that

$$\mathsf{Var}(X_j) = \mathsf{Var}(\frac{1}{p} {\delta_1}^{(\bar{\boldsymbol{c}})} \cdot \frac{1}{p} {\delta_1}^{(\boldsymbol{c})}|_j) = n V_{\frac{1}{p}{\delta_1}^{(\bar{\boldsymbol{c}})}} V_{\frac{1}{p}{\delta_1}^{(\boldsymbol{c})}},$$

which is equivalent to prove that

$$\sum_{j_1 \neq j_2} \mathsf{Cov}({\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_1} {\delta_1}^{(\boldsymbol{c})}|_{i-j_1}, {\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_2} {\delta_1}^{(\boldsymbol{c})}|_{i-j_2}) = 0,$$

i.e.,

$$\mathbb{E}[{\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_1} {\delta_1}^{(\boldsymbol{c})}|_{i-j_1} {\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_2} {\delta_1}^{(\boldsymbol{c})}|_{i-j_2}] = 0 \quad \text{for } j_1 \neq j_2.$$

Using the tower property, we can write

$$\mathbb{E}[{\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_1} {\delta_1}^{(\boldsymbol{c})}|_{i-j_1} {\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_2} {\delta_1}^{(\boldsymbol{c})}|_{i-j_2}] = \mathbb{E}\Big[{\delta_1}^{(\boldsymbol{c})}|_{i-j_1} {\delta_1}^{(\boldsymbol{c})}|_{i-j_2} \mathbb{E}[{\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_2} {\delta_1}^{(\bar{\boldsymbol{c}})}|_{j_1} \mid \boldsymbol{c}]\Big].$$

Focusing on

$$\delta_1{}^{(\bar{\boldsymbol{c}})} = t[-t^{-1}\sigma_k(c_1)a_{\mathsf{ks}}]_p,$$

we first observe that, conditionally on $\boldsymbol{c}$ the element $\sigma_k(c_1)$ is fixed, while $a_{\mathsf{ks}} \leftarrow \mathcal{U}_{pq}$ is freshly generated during the key switching procedure. For a fixed $\sigma_k(c_1)$, multiplication by $\sigma_k(c_1)$ is an $\mathcal{R}_p$-linear map

$$m_{\sigma_k(c_1)} : \mathcal{R}_p \longrightarrow \mathcal{R}_p, \qquad x \longmapsto \sigma_k(c_1)x.$$

Hence, since $a_{\mathsf{ks}} \mod p \leftarrow \mathcal{U}_p$, $\sigma_k(c_1) \cdot a_{\mathsf{ks}}$ is uniform over the image of $m_{\sigma_k(c_1)}$, that coincides with the principal ideal $J \coloneqq \sigma_k(c_1)\,\mathcal{R}_p$. Fix $j_1 \neq j_2$ and let $\pi : \mathcal{R}_p \to \mathbb{F}_p^2$ be the projection onto the coordinates j_1, j_2. The composition $\pi \circ m_{\sigma_k(c_1)}$ is $\mathbb{F}_p$-linear, hence a homomorphism of finite abelian groups. Using Lemma 4 we know that for every parameter set considered in this work $\pi(J) = \mathbb{F}_p^2$ with overwhelming probability. Indeed, $\dim_{\mathbb{F}_p} J = n - \deg g$ where $g \coloneqq \gcd(\sigma_k(c_1), x^n + 1)$. Hence,

$$\dim_{\mathbb{F}_p} J > \frac{n}{2} \Leftrightarrow \deg g < \frac{n}{2}$$

However,

$$\mathbb{P}[\deg g \geq \frac{n}{2}] \leq \sum_{\substack{g \mid x^n+1 \\ \deg g \geq n/2}} \mathbb{P}[g \mid \sigma_k(c_1)] = \sum_{\substack{g \mid x^n+1 \\ \deg g \geq n/2}} p^{-\deg g} \leq 2^n p^{-n/2} = 2^{-n/2(\log_2 p - 2)},$$

which is negligible with the common choice of parameters. This means that we can apply Lemma 4 and assume the two coefficients $\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_2}, \delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_1}$ to be independent and uniform over $\mathbb{F}_p$, i.e.,

$$\mathbb{E}[\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_2}\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_1} \mid \boldsymbol{c}] = \mathbb{E}[\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_2}\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_1}] = \mathbb{E}[\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_2}]\mathbb{E}[\delta_1{}^{(\bar{\boldsymbol{c}})}|_{j_1}] = 0,$$

thus concluding the proof.

Remark 2. The results presented so far are based on the assumption that the ciphertext moduli ratio is large enough to ensure that Lemma 3 holds. However, for parameter selection, it is necessary to quantify how large this ratio actually needs to be. Let ν be the initial critical quantity and $\nu_{\mathsf{ms}} = \frac{q'_\ell}{q}\big(\nu + \delta_1 s\big)$ the corresponding quantity after modulus switching. According to [3], we define a parameter α such that $\mathsf{Var}(\nu/p|_i) < \alpha\mathsf{Var}(\delta_1 s/p|_i)$, where we set $\alpha = 1/100$. In this way, we ensure that the error introduced by modulus switch dominates the one derived from the previous error term, while still imposing conditions on the ratio of the moduli that are not too restrictive and allow us to provide an improvement in the choice of the parameters.

As emphasized throughout this work, avoiding underestimation is essential to ensure the correct functioning of the scheme and preserve its security. Since our estimates are very close to the experimental values, we suggest to incorporate the margin factor $\epsilon = 6\alpha$ in the variance estimates of Theorems 2 to 4, to account

for the residual contribution of the pre-modulus-switching noise. Hence, the variance becomes $(4+\epsilon)nV_{\mathsf{ms}}^2$ (Theorems 2 and 3) and $(2+\epsilon)nV_{\mathsf{ms}}^2$ (Theorem 4). The rationale behind this correction is explained in depth in [3]. However, the idea is the following. After modulus switching, ciphertext multiplication involves products of critical quantities of the form $\nu_{\mathsf{ms}} = \frac{q'}{q}(\nu + \delta s)$. In the worst-case scenario, where the two quantities are equal (maximum dependency) one must also account for the cross terms between ν and δs. Moreover, since modulus switching is applied before each multiplication step, the highest power of s appearing in ν can be assumed to be at most quadratic, hence it is sufficient to use as correction factor $F(2,1)$ [3, Definition 4], which leads precisely to the given choice of ϵ. We observe that, for $\alpha = 1/100$, the induced correction is very small.

5 Variance Estimates for Polynomial Evaluation

The results presented in Section 4 provide the building blocks for estimating the noise growth in the evaluation of an arbitrary polynomial over a ciphertext. The resulting variance depends on the evaluation strategy, since different strategies may produce different multiplicative depths and apply modulus switching at different stages.

A key insight of our work is that the variance of the coefficients after polynomial evaluation is ultimately determined by (i) the multiplicative depth associated with the computation of each power; (ii) the terms that, in the final addition, do not undergo a final modulus switch (see Proposition 3). These two factors determine which variance estimates from Section 4 contribute to the final result.

This observation provides a general methodology for analyzing polynomial evaluation strategies: once the sequence of homomorphic operations and modulus-switching steps is specified, the variance can be estimated by tracking the contributions of the individual terms and their final addition. In the following, we illustrate this methodology through two standard strategies, square-and-multiply and baby-step giant-step, and discuss its applicability to rotation-based methods, which additionally involve products of a ciphertext with its rotations.

5.1 Square and Multiply

The following results provide a concrete instantiation of the method presented so far, giving an accurate estimate of the noise derived from the evaluation of a polynomial on a ciphertext, using the square-and-multiply algorithm. We chose to compute each power of a ciphertext $\boldsymbol{c}$ using the *binary method* [27], which means that we apply recursively

$$\boldsymbol{c}^j = \boldsymbol{c}^{2^{\lfloor \log_2 j \rfloor}} \boldsymbol{c}^{j-2^{\lfloor \log_2 j \rfloor}}.$$

In this way, monomials whose exponents are powers of 2 are obtained by multiplying a ciphertext by itself, while other powers are given by the product of two

distinct powers. Estimates of the critical quantities associated with these computations are covered by Theorems 2 and 3, respectively. Note that the result in Theorem 5 holds unchanged when powers are computed using *balanced* binary multiplication trees.

Theorem 5. *Let $(\boldsymbol{c}, q, \nu)$ be an extended ciphertext and let $f(x) = \sum_{j=1}^{\kappa} a_j x^j$ be a polynomial, with $\kappa \geq 2$ and $d = \lceil \log_2 \kappa \rceil$. Let $\boldsymbol{c}_f$ denote the ciphertext obtained by evaluating f on $\boldsymbol{c}$ and let ν_f be the associated critical quantity. Then, using the square-and-multiply algorithm, we have*

$$\mathsf{Var}(\nu_f|_i) \leq m V_{\mathsf{ms}} + \sum_{\lceil \log_2 j \rceil = d} 4 n a_j^2 V_{\mathsf{ms}}^2.$$

where m is the number of monomials of $f(x)$ of degree at most 2^{d-1}, namely, $m = \#\{1 \leq j \leq 2^{d-1} : \ a_j \neq 0\}$.

Proof. For a fixed power j, consider the corresponding monomial of f evaluated in $\boldsymbol{c}$, namely $a_j \boldsymbol{c}^j$. Then,

- if j is a power of 2, then $\boldsymbol{c}^j = \boldsymbol{c}^{j/2} \boldsymbol{c}^{j/2}$, thus the coefficient variance of the associated critical quantity is approximately $4 n V_{\mathsf{ms}}^2$ by Theorem 2.
- if j is not a power of 2, then $\boldsymbol{c}^j = \boldsymbol{c}^k \boldsymbol{c}^l$, with $k \neq l$, hence the two factors are distinct powers of $\boldsymbol{c}$. Thus, the coefficient variance of the associated critical quantity is upper bounded by $4 n V_{\mathsf{ms}}^2$ according to Theorem 3.

At this point, to perform the addition, the ciphertexts must lie in the same ring. Hence,

- if $\lceil \log_2 j \rceil < d = \lceil \log_2 \kappa \rceil$, i.e. $j \leq 2^{d-1}$, modulus switching is applied to $a_j \boldsymbol{c}^j$, yielding an associated critical quantity equal to ν_{ms} with coefficient variance V_{ms} (Lemma 3).
- if $\lceil \log_2 j \rceil = d$, modulus switching is not applied to the ciphertext before the addition, hence the contribution of the ciphertext to the final variance is upper bounded by $4 n a_j^2 V_{\mathsf{ms}}^2$, using the fact that $V_{\gamma c} = \gamma^2 V_c$, $\forall \gamma \in \mathbb{Z}, c \in \mathcal{R}$.

The claim follows straightforwardly by summing all ciphertexts using Proposition 3 and observing that the number of ciphertexts that are modulus switched before addition is exactly m. □

A simplified estimate is provided in the following corollary.

Corollary 1. *Let $(\boldsymbol{c}, q, \nu)$ be an extended ciphertext and let $f(x) = \sum_{j=1}^{\kappa} a_j x^j$ be a polynomial, with $\kappa \geq 2$ and $d = \lceil \log_2 \kappa \rceil$. Let $\boldsymbol{c}_f$ denote the ciphertext obtained by evaluating f on $\boldsymbol{c}$ and let ν_f be the associated critical quantity. Then,*

$$\mathsf{Var}(\nu_f|_i) \leq 2^{d-1} V_{\mathsf{ms}} + \sum_{\lceil \log_2 j \rceil = d} 4 n a_j^2 V_{\mathsf{ms}}^2 \approx 4 n V_{\mathsf{ms}}^2 \sum_{\lceil \log_2 j \rceil = d} a_j^2.$$

Proof. The claim follows immediately by Theorem 5, by observing that: (i) m is by definition at most 2^{d-1}, (ii) in the concrete usage of BGV, $n \geq 2^{12}$, hence, the first term is negligible with respect to the second. □

In other words, Corollary 1 shows that only the monomials not refreshed by modulus switching contribute to the final variance estimate, namely those requiring the maximum multiplicative depth among the terms of f. For example, when evaluating $x^8 + x^4$, only the contribution of x^8 appears in the estimate, since the ciphertext associated with x^4 must undergo an additional modulus-switching before being added to x^8. In contrast, for $x^8 + x^5$, both terms contribute to the final variance.

5.2 Baby-Step Giant-Step

The next statement gives an instantiation of our method when the baby-step giant-step algorithm [33] is employed. Let $f(x) = \sum_{j=1}^{\kappa} a_j x^j$ be a polynomial of degree $\kappa \geq 2$, and set $m = \lceil \sqrt{\kappa + 1} \rceil$. The polynomial can be decomposed as

$$f(x) = \sum_{j=0}^{\lfloor \kappa/m \rfloor} (x^m)^j B_j(x), \qquad B_j(x) = \sum_{i=0}^{m-1} a_{jm+i} x^i, \tag{8}$$

following [32, Section 4]. In our analysis, each giant power $(\boldsymbol{c}^m)^j$ is computed using a balanced binary multiplication tree, rather than combining the blocks as in [38]. It is worth noting that, even for a fixed polynomial algorithm, the noise estimate also depends on implementation choices such as the order of the homomorphic operations and when modulus switching is applied throughout the computation.

Theorem 6. *Let $(\boldsymbol{c}, q, \nu)$ be an extended ciphertext and let $f(x) = \sum_{j=1}^{\kappa} a_j x^j$ be a polynomial of degree $\kappa \geq 2$. Set*

$$m = \lceil \sqrt{\kappa + 1} \rceil, \qquad \kappa = \rho m + \beta, \qquad 0 \leq \beta < m.$$

Let $\boldsymbol{c}_f$ be the ciphertext obtained by evaluating f on $\boldsymbol{c}$ using the baby-step giant-step algorithm described above, and let ν_f denote its associated critical quantity. Finally, define

$$A = \begin{cases} 2^{\lceil \log_2 \rho \rceil - 1}, & \text{if } \rho \geq 2, \\ 0, & \text{if } \rho = 1. \end{cases}$$

If $\beta \neq 0$, then

$$\mathsf{Var}(\nu_f|_i) \leq (\rho + 1) V_{\mathsf{ms}} + 4n(\rho - A) V_{\mathsf{ms}}^2 .$$

If $\beta = 0$, then

$$\mathsf{Var}(\nu_f|_i) \leq \begin{cases} V_{\mathsf{ms}} + 4n\, a_\kappa^2\, V_{\mathsf{ms}}^2, & \rho = 1, \\ 2V_{\mathsf{ms}} + 4n\big(1 + a_\kappa^2\big) V_{\mathsf{ms}}^2, & \rho = 2, \\ (\rho + 1) V_{\mathsf{ms}} + 4n(\rho - A - 1) V_{\mathsf{ms}}^2, & \rho \geq 3 \text{ and } (\star), \\ (A + 1) V_{\mathsf{ms}} + 2n\Big(A + 2 \sum\limits_{A < j \leq \rho} a_{jm}^2\Big) V_{\mathsf{ms}}^2, & \text{otherwise}, \end{cases}$$

where $(\star)$ *indicates that there exists* j *with* $(A+1)m < j < \kappa$, $m \nmid j$ *and* $a_j \neq 0$.

Sketch of the proof. First, we write the polynomial as in Equation (8), that is $f(x) = \sum_{j=0}^{\rho} x^{jm} B_j(x)$ and we evaluate each block separately. For a block with $j \geq 2$, the degrees appearing in B_j are smaller than m, meaning that the corresponding baby-step powers must undergo modulus switch before being multiplied by the giant power $\boldsymbol{c}^{jm}$. Hence, each nonzero term in the block has a variance contribution bounded by V_{ms}, while the multiplication by the giant power contributes at most $4nV_{\mathsf{ms}}^2$. Finally, when all blocks are added, those below the maximum multiplicative level undergo an additional modulus switching, and contribute at most V_{ms} each, while blocks at maximum level retain the contribution of their products. Depending on their structure, this contribution is bounded by $4nV_{\mathsf{ms}}^2$ or, when a block reduces to a single monomial $a_{jm}x^{jm}$, by $4na_{jm}^2 V_{\mathsf{ms}}^2$ (which can only occur if $\beta = 0$). The blocks B_0 and B_1 require separate consideration, since they do not follow the same pattern as the remaining blocks. In particular, the resulting bound depends on whether $\beta = 0$ and on which coefficients of f are nonzero. Counting the contributions at the maximal level concludes the proof. The complete proof is provided in Appendix B. □

Corollary 2. *Let* $(\boldsymbol{c}, q, \nu)$ *be an extended ciphertext, and let* $f(x) = \sum_{j=1}^{\kappa} a_j x^j$ *be a polynomial of degree* $\kappa \geq 2$, *where* $\kappa = \rho m + \beta$, $m = \lceil \sqrt{\kappa + 1} \rceil$ *and* $0 \leq \beta < m$. *Let* $\boldsymbol{c}_f$ *be the ciphertext obtained by evaluating* f *on* $\boldsymbol{c}$ *using the baby-step giant-step algorithm, and let* ν_f *denote its critical quantity. Then,*

$$\mathsf{Var}(\nu_f|_i) \leq (\rho+1)V_{\mathsf{ms}} + 4n\Big(\left\lceil \frac{\rho}{2} \right\rceil + \delta \sum_{A<j\leq\rho} a_{jm}^2\Big)V_{\mathsf{ms}}^2,$$

where $\delta = 1$ *if* $\beta = 0$ *and* 0 *otherwise.*

Proof. If $\beta \neq 0$, the bound in Theorem 6 can be directly upper bounded using that $\rho - A \leq \lceil \rho/2 \rceil$. If $\beta = 0$, the first two cases of Theorem 6 are covered by the claimed bound. The case $\rho \geq 3$ holds since $\rho - A - 1 \leq \lceil \rho/2 \rceil$, while for the remaining case, we can use that

$$A + 2\sum_{A<j\leq\rho} a_{jm}^2 \leq 2\left(\left\lceil \frac{\rho}{2} \right\rceil + \sum_{A<j\leq\rho} a_{jm}^2\right).$$

□

Corollary 2 provides a simpler and more compact bound, which is close to Theorem 6 when $\beta \neq 0$ or when the coefficients a_{jm}, $A < j \leq \rho$, are small. However, if $\beta = 0$ and some of these coefficients are large, it may considerably overestimate the variance: for instance, for $f(x) = 128x^{72} + x^{64}$ the gap is about 12 bits, since in this case the coefficient a_{72} is absorbed by modulus switching. In such cases Theorem 6 should be preferred.

Choosing the algorithm. A further consequence of our framework is that it provides a systematic way to compare different polynomial-evaluation strategies in BGV. Indeed, once the sequence of homomorphic operations and modulus-switching steps is fixed, our results allow one to estimate the corresponding noise growth and, consequently, to choose the evaluation strategy that optimizes performances. For square-and-multiply (SM) and baby-step giant-step (BSGS), the differences can be summarized as follows.

- *Depth.* SM achieves the minimal depth, in fact BSGS generally requires one or two additional levels.
- *Noise.* In SM, the final variance grows with a_j^2 for the monomials of maximal depth, whereas in BSGS the coefficients are absorbed by modulus switching (Theorem 6). This advantage, however, comes from applying the coefficients *before* the last multiplication, and the same can be done in SM by scaling one of the two factors before computing the final product of each maximal-depth monomial. In this way, the coefficient contribution is absorbed by the subsequent modulus switching step, without increasing the multiplicative depth.
- *Number of multiplications.* For a dense polynomial of degree κ, SM requires $\kappa - 1$ ciphertext multiplications, and BSGS about $3\sqrt{\kappa}$. On the other hand, for a sparse polynomial, SM only computes the powers that are needed.

Therefore, SM is preferable when minimizing the multiplicative depth, and hence the ciphertext modulus, is the main objective, as well as for sufficiently sparse polynomials. Conversely, BSGS is advantageous for large dense polynomials when reducing the number of ciphertext multiplications is the primary goal and one or two additional levels are acceptable.

5.3 Rotation-Based Polynomial Evaluation

Polynomial evaluation methods considered so far rely only on additions and multiplications. Recently, the authors of [32] showed that, for BGV and BFV, the number of ciphertext-ciphertext multiplications required to evaluate a fixed polynomial can be reduced by exploiting the Galois structure of plaintext slots of high algebraic rank.

Intuitively, the main idea is to express a polynomial in terms of the algebraic norm or trace. To ease the explanation, we focus on the first, which is also the one implemented by the authors in the Microsoft SEAL library [35]. According to this method, under [32, Heuristic 4.2.] it is possible to write a polynomial f of degree smaller than the rank d of the plaintext slots as

$$f(x) = \mathrm{N}(\theta - x) - x^d - g(x), \tag{9}$$

for a suitable slot element θ and a polynomial g, which can be precomputed at the beginning. As explained in [32, Section 4.1] computing $\mathrm{N}(\theta - x)$ consists in $\log_2(d)$ multiplications of a ciphertext by a rotation of itself. The critical quantity

resulting from this computation is modeled by our work in Section 4.3, whereas the computations of x^d and $g(x)$ are well studied by the methods proposed above.

Another more recent work [39] extends this approach to more general slot structures and to polynomials whose degree exceeds the slot rank. Although the corresponding algorithms are different, they are built from the same basic operations considered in our analysis: squaring, product of possibly dependent ciphertexts, product involving rotations and addition of possibly dependent ciphertexts. Consequently, once the multiplicative depth of the intermediate ciphertexts and the levels at which the final additions are performed are identified, the variance estimates follow from Section 4, in the same way as for the other polynomial evaluation strategies considered in this section.

This further highlights that the framework developed in this paper is not tied to any specific polynomial evaluation strategy. Rather, it provides a general methodology that can cover not only classic polynomial evaluation methods but also more innovative techniques.

6 Experimental Validation

In this section, we validate our theoretical results by comparing the estimates of the variance obtained through our approach with empirical observations (Section 6.1). Experimental evaluations of the variance were carried out using the OpenFHE library [1]. Note that OpenFHE does not provide dedicated functions for computing powers of a ciphertext or for evaluating polynomial functions on ciphertexts. Therefore, these operations must be implemented manually using the `EvalMult` and `EvalAdd` functions. In our implementation, ciphertext powers are computed using both the square-and-multiply and the baby-step giant-step algorithms.

We also provide our own selection of the moduli based on our variance estimation method (Section 6.2). We compare our parameters with those employed in OpenFHE and HElib [36], showing that our approach allows us to use significantly smaller moduli, leading to considerable efficiency gains without compromising security. Indeed, because our variance estimation never underestimates the actual variance, the decryption failure probability is negligible[4].

The code used for the experiments is publicly available at
https://github.com/nadirmur/openFHE

6.1 Variance estimation

The experiments in this section were performed for different values of the ring dimension n and the plaintext modulus is set to $t = 65537$. For each value of n, the experimental variance was estimated by sampling 10000 different ciphertexts. For simplicity, we report the base-2 logarithm of the variance. In all the

[4] Note that the decryption failure probability is never exactly zero, as average-case analysis intrinsically relies on probabilistic bounds.

theoretical estimates reported below, we incorporate the correction introduced in Remark 2, setting $\alpha = 1/100$ and $\epsilon = 6\alpha$. In particular, contributions of the form $4nV_{\text{ms}}^2$ and $2nV_{\text{ms}}^2$ are replaced by $(4+\epsilon)nV_{\text{ms}}^2$ and $(2+\epsilon)nV_{\text{ms}}^2$, respectively.

Given a ciphertext $\boldsymbol{c}$, Figures 1a and 1b report our estimate of the variance of the first coefficient of the critical quantity associated with $\boldsymbol{c}^2$ and $\boldsymbol{c}^7$, which can be computed according to Theorems 2 and 3, respectively. Figures 1c and 1d analyze the variance of the first coefficient of the critical quantity associated with the product of a ciphertext by a rotation of itself, according to Theorem 4. Specifically, we consider σ_5 and σ_{n-1} as instances.

n	2^{13}	2^{14}	2^{15}
exp	96.61	99.65	102.65
our	96.68	99.68	102.68

(a) Variances for $\boldsymbol{c}^2$.

n	2^{13}	2^{14}	2^{15}
exp	95.64	98.66	101.49
our	96.68	99.68	102.68

(b) Variances for $\boldsymbol{c}^7$.

n	2^{13}	2^{14}	2^{15}
exp	95.65	98.68	101.67
our	95.70	98.70	101.70

(c) Variances for ν_{σ_5}.

n	2^{13}	2^{14}	2^{15}
exp	95.65	98.65	101.65
our	95.70	98.70	101.70

(d) Variances for $\nu_{\sigma_{n-1}}$.

Fig. 1: Comparison between our theoretical estimates and the empirical ones.

In Figure 2, given a ciphertext $\boldsymbol{c}$, we report our (*our*) estimate of the variance of the first coefficient of the critical quantity associated with two polynomials: $T_5(x) = 16x^5 - 20x^3 + 5x$ and $g(x) = x^8 + x^7 + x^6 + x^5$, considering Square and Multiply (SM) and Baby-Step Giant-Step (BSGS) algorithms. Moreover, the tag "exp" shows the observed values from OpenFHE [1] library with 10000 polynomial samples setting the parameters according to the required multiplicative depth, with $t = 65537$, $\chi_s = \mathcal{U}_3$, $\chi_e = \mathcal{DG}(0, \sigma^2)$, with $\sigma = 3.19$ and security level at least 128-bit.

n	2^{13}	2^{14}	2^{15}
exp (SM)	103.66	106.69	109.66
our (Theorem 5)	104.68	107.68	110.68
exp (BSGS)	95.70	98.62	101.67
our (Theorem 6)	96.68	99.68	102.68

(a) Variances for $T_5(\boldsymbol{c})$

n	2^{13}	2^{14}	2^{15}
	97.98	101.02	104.00
	98.68	101.68	104.68
	95.64	98.65	101.70
	96.68	99.68	102.68

(b) Variances for $g(\boldsymbol{c})$

Fig. 2: Comparison between our theoretical estimates and the empirical variances.

Moreover, to highlight how the choice between the square-and-multiply and baby-step giant-step algorithms affects noise growth, we consider the following two polynomials $h_1(x) = 10x^{20} + 10x^{16} + 10x^3$ and $h_2(x) = 10x^{20} + 10x^{15}$. This specific example is also interesting since one might expect $h_1(x)$ to produce more noise than $h_2(x)$, since it contains more monomials and its highest-degree monomial other than x^{20} has a larger degree. However, as Figure 3 shows, this is not the case.

n	2^{13}	2^{14}	2^{15}
exp (SM)	102.30	105.27	108.28
our (Theorem 5)	103.33	106.33	109.33
exp (BSGS)	95.64	98.66	101.64
our (Theorem 6)	96.68	99.68	102.68

(a) Variances for $h_1(\boldsymbol{c})$

n	2^{13}	2^{14}	2^{15}
	102.30	105.27	108.28
	103.33	106.33	109.33
	103.87	106.93	109.89
	104.33	107.33	110.33

(b) Variances for $h_2(\boldsymbol{c})$

Fig. 3: Comparison between our theoretical and the empirical variances.

As shown in the tables, our method provides very accurate estimates of the variance with differences from the experimental values below 1.2 bits. Moreover, this high level of accuracy is achieved without ever underestimating it, which is essential for ensuring both safety and correct system operation.

6.2 Selection of Ciphertext Moduli

Section 4 provides accurate estimates for the variance of the coefficients of the critical quantity after the main homomorphic operations. Combined with the fact that such coefficients can be accurately modeled by Gaussian distributions[5], these estimates allow us to derive tight probabilistic bounds on the error norm. Consequently, it is possible to determine suitable conditions on the ciphertext moduli to guarantee the correct functioning of the scheme.

Indeed, if the coefficients of the critical quantity ν are modeled as Gaussian random variables with variance V, then the following bound on the decryption failure probability holds

$$\mathbb{P}\left(|\nu|_\infty > \frac{q}{2}\right) \leq n\left(1 - \operatorname{erf}\left(\frac{q}{2\sqrt{2V}}\right)\right).$$

Hence, by defining a *security parameter* D such that $D \leq q/2\sqrt{2V}$, it follows that

$$\mathbb{P}\left(\|\nu\|_\infty > \frac{q}{2}\right) \leq n\left(1 - \operatorname{erf}(D)\right),$$

which means that fixing the modulus $q \geq 2D\sqrt{2V}$, we can properly select D to guarantee that the probability of decryption failure remains negligibly small.

[5] We refer to [3] for a detailed discussion on this assumption.

A common choice for practical usage is $D = 8$ since, for instance, it results in decryption failure probability of approximately 2^{-83} when $n = 2^{13}$.

On the Gaussian Assumption. As widely discussed in [3], we can model the coefficients of the critical quantities involved in any circuit of operations as Gaussian thanks to the systematic usage of modulus switch. Specifically, the authors of [3] show that it is necessary to impose a lower bound on the ciphertext moduli ratio p, that is

$$p > \sqrt{\frac{V}{\alpha V_{\mathsf{ms}}}} \tag{10}$$

where V denotes the variance before modulus switch is computed, V_{ms} is as in Equation (4) and α is a parameter which is heuristically selected as $\alpha = 1/100$. Note that the primes currently adopted by most existing libraries already satisfy this requirement. Hence, this does not compromise the modulus-size improvement proposed in this work compared to the state-of-the-art.

Comparison with OpenFHE. Given a multiplicative depth M, OpenFHE uses $M + 2$ moduli in order to allow an additional modulus switching after the final multiplication. In our method, the primes $p_0, \ldots, p_{M+1}$ whose product provides the ciphertext modulus q are selected imposing the condition (10), namely, for $0 < \ell < M + 1$, $p_\ell > \sqrt{V/\alpha V_{\mathsf{ms}}}$, where V is our variance estimate and $\alpha = 1/100$. The primes p_0 and p_{M+1} must satisfy $p_0 > 2D\sqrt{2V_{\mathsf{ms}}}$, and $p_{M+1} > \sqrt{V_{\mathsf{clean}}/\alpha V_{\mathsf{ms}}}$, where V_{clean} is the variance of a fresh ciphertext. In the cases $n = 2^{13}, 2^{14}, 2^{15}$, the variance of a fresh ciphertext is $48.76, 49.76, 50.76$, respectively (see [3]). For a fair comparison, each prime is chosen as the smallest NTT-friendly prime, satisfying the corresponding lower bound, with all primes distinct and different from the plaintext modulus.

Observe that when selecting the moduli in this case, not only the multiplicative depth must be taken into account, but also the number of additions required to perform ciphertext-scalar multiplications. In OpenFHE, this parameter is specified through `SetEvalAddCount`. For example, in the evaluation of the Chebyshev polynomial, the required number of additions is equal to 16, while the multiplicative depth is 3.

In Figure 4, we compare the ciphertext modulus bit sizes selected by OpenFHE with those obtained using our variance estimation method when $\boldsymbol{c}^2$ and $\boldsymbol{c}^7$ are computed.

n	2^{13}	2^{14}	2^{15}
OpenFHE	82.8	85.3	88.6
Our method	73.0	75.0	78.8

(a) Modulus bit sizes for computing $\boldsymbol{c}^2$ (multiplicative depth 1).

n	2^{13}	2^{14}	2^{15}
OpenFHE	154.8	159.3	164.6
Our method	135.5	139.5	145.3

(b) Modulus bit sizes for computing $\boldsymbol{c}^7$ (multiplicative depth 3).

Fig. 4: Comparison of ciphertext modulus bit sizes between OpenFHE (v1.5.1) and our selection method.

In particular, Figure 1 shows that our estimates are very tight with respect to the actual variance of the critical quantity, while never underestimating it in any of our experiments. As a result, our approach enables a more efficient modulus selection, minimizing the modulus size without compromising correctness and security, as the probability of decryption failure remains negligible. Similar results can be observed in Figure 5 for the moduli size for computing the polynomials $T_5(\boldsymbol{c})$ and $g(\boldsymbol{c})$, using the square and multiply method (Section 5.1).

n	2^{13}	2^{14}	2^{15}
OpenFHE	160.8	165.3	170.6
Our method	143.8	147.8	153.6

(a) Modulus bit sizes for computing $T_5(\boldsymbol{c})$ (multiplicative depth 3 and addition count 16).

n	2^{13}	2^{14}	2^{15}
OpenFHE	154.8	159.3	164.6
Our method	136.5	140.5	146.3

(b) Modulus bit sizes for computing $g(\boldsymbol{c})$ (multiplicative depth 3 and addition count 3).

Fig. 5: Comparison of ciphertext modulus bit sizes between OpenFHE (v1.5.1) and our selection method.

Comparison with HElib. Here we provide a comparison of our moduli selection with respect to those currently employed in HElib [24]. Note that in [16] a strategy for optimizing HElib parameters is proposed. However, we do not provide a direct comparison with this approach, since it is tailored to the specific library and optimizes a different objective, namely the trade-off involving the special modulus used for key switching, which we deliberately do not model. For reference, [16, Table 8] results in a modulus ratio of 32/33/34 bits (under the most permissive setting), while our framework gives 31/32/33 bits, for $n = 2^{13}, 2^{14}, 2^{15}$. However, since both the optimization objectives and parameter settings differ, these results should not be interpreted as a direct comparison. Indeed, in this work, we intend to provide a general framework for optimizing the ciphertext moduli selection that is library independent.

Unlike OpenFHE, the latest version of HElib does not select the moduli dimension based on the multiplicative depth of a circuit. Instead, the parameter selection is based on the bit length of the largest modulus in the chain. Hence, to have a significant comparison, we follow the same strategy of [16] and compare the ratio between the moduli of successive levels. In HElib, by construction, this ratio is always larger than 36 bits. However, as pointed out in [16], in practical usage it is always much larger, with size of approximately 54 bits on average, for ring dimensions $n \in \{2^{13}, 2^{14}, 2^{15}\}$. Instead, in our method, we only require that the primes p_ℓ, for $0 < \ell < M+1$ (M multiplicative depth), satisfy eq. (10), namely $p_\ell > \sqrt{V/\alpha V_{\mathsf{ms}}}$, where V denotes the variance estimate obtained from our bounds prior to modulus switching and is determined by the structure of the evaluating circuit. As an illustrative example, we consider a circuit where,

at each level, a multiplication between two dependent ciphertexts occurs. In this context, Theorem 2 and Remark 2 yield the estimate $V \approx (4+\epsilon)nV_{\mathsf{ms}}^2$, leading to the constraint

$$p_\ell > \sqrt{\frac{(4+\epsilon)nV_{\mathsf{ms}}}{\alpha}}.$$

We use this scenario as a reference setting for the comparisons presented in Table 1. Different circuits may yield distinct values of V, depending on the sequence of homomorphic operations performed. However, our analysis offers a systematic approach to accurately estimate the corresponding variance in each case and to derive the associated modulus constraints.

In Table 1 we present the bit-size of the ratio between adjacent moduli resulting from our approach, for $n \in \{2^{13}, 2^{14}, 2^{15}\}$. The experiments assume ternary secret keys with Hamming weight $h = n/2$, in accordance with the library specifications, and plaintext modulus $t = 65537$.

n	2^{13}	2^{14}	2^{15}
$\log_2(p_\ell)$	31.04	32.04	33.04

Table 1: Ratio between adjacent ciphertext moduli for different ring dimensions n, according to our approach, represented using their bit size.

The results show that our approach consistently produces substantially smaller modulus ratios, achieving reductions of up to 5 bits between consecutive ciphertext moduli compared to the smallest possible ratio dimension in HElib. The improvement becomes considerably more significant when compared with the modulus ratios typically adopted in practical deployments, where the reduction can reach 23 bits per level.

7 Conclusions

This work provides the first average-case analysis for the BGV scheme that addresses the case of ciphertext dependencies. We show that such dependencies have a non-negligible impact on homomorphic computations, as they increase the expected variance of the error coefficients involved in operations between ciphertexts. At the same time, we show that these effects can be effectively controlled in practice, thanks to modulus switching, which helps attenuate the contribution of dependency-induced correlations.

We derive accurate variance estimates for homomorphic additions and multiplications involving dependent ciphertexts, accounting not just for secret/public-key-induced correlations but also for those induced by ciphertext itself. In particular, we extend our analysis to the homomorphic evaluation of polynomial

functions, a central component of practical FHE applications. This allows us to obtain tight bounds on the resulting error growth.

Experimental findings indicate that the proposed estimates align closely with observed variances, with discrepancies between theoretical and experimental values never exceeding 1.2 bits and consistently avoiding underestimation.

Finally, these estimates are leveraged for ciphertext modulus selection, leading to significantly smaller parameters and improved efficiency compared to those currently adopted in state-of-the-art FHE libraries. Specifically, in comparison with OpenFHE, the methodology reduces the ciphertext modulus by an average of approximately 16.3 bits across all evaluated benchmarks, corresponding to an average reduction of about 11.5%. Additionally, relative to the modulus ratios currently employed in HElib, the approach can decrease the size of the ratios between consecutive moduli by 23 bits per level in practical parameter settings.

Future work could investigate the applicability of the proposed framework to other RLWE-based FHE schemes. For example, we believe that for the BFV scheme, an adaptation may require a more complicated investigation, due to the different role played by the modulus switching technique.

Acknowledgment

The second and third authors were partially supported by the Italian Ministry of University and Research in the framework of the Call for Proposals for scrolling of final rankings of the PRIN 2022 call - Protocol no. 2022RFAZCJ.

References

1. Al Badawi, A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., et al.: OpenFHE: Open-Source Fully Homomorphic Encryption Library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. pp. 53–63 (2022)
2. Biasioli, B., Marcolla, C., Calderini, M., Mono, J.: Improving and automating BFV parameters selection: An average-case approach. Cryptology ePrint Archive, Paper 2023/600 (2023), https://eprint.iacr.org/2023/600
3. Biasioli, B., Marcolla, C., Murru, N., Urani, M.: Accurate BGV parameters selection: Accounting for secret and public key dependencies in average-case analysis. Cryptology ePrint Archive, Paper 2025/2027 (2025), https://eprint.iacr.org/2025/2027
4. Bossuat, J., Costache, A., Mouchet, C., Nürnberger, L., Troncoso-Pastoriza, J.R.: Accurate and composable noise estimates for CKKS with application to exact HE computation. IACR Commun. Cryptol. **2**(2), 8 (2025)
5. Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Advances in Cryptology – CRYPTO 2012. pp. 868–886 (2012)
6. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)

7. Checri, M., Sirdey, R., Boudguiga, A., Bultel, J.P.: On the practical cpad security of "exact" and threshold fhe schemes and libraries. In: Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part III. p. 3–33. Springer-Verlag, Berlin, Heidelberg (2024)
8. Cheon, J.H., Choe, H., Passelègue, A., Stehlé, D., Suvanto, E.: Attacks against the ind-cpad security of exact fhe schemes. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. p. 2505–2519. CCS '24, Association for Computing Machinery, New York, NY, USA (2024)
9. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A full RNS variant of approximate homomorphic encryption. In: International Conference on Selected Areas in Cryptography – SAC 2018. pp. 347–368. Springer (2018)
10. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Advances in Cryptology – ASIACRYPT 2017. pp. 409–437 (2017)
11. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: Advances in Cryptology – ASIACRYPT 2016. pp. 3–33 (2016)
12. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: Fast Fully Homomorphic Encryption over the Torus. Journal of Cryptology **33**(1), 34–91 (2020)
13. Costache, A., Smart, N.P.: Which ring based somewhat homomorphic encryption scheme is best? In: Topics in Cryptology – CT-RSA 2016. pp. 325–340 (2016)
14. Costache, A., Curtis, B.R., Hales, E., Murphy, S., Ogilvie, T., Player, R.: On the precision loss in approximate homomorphic encryption. In: International Conference on Selected Areas in Cryptography - SAC 2023. pp. 325–345. Springer (2023)
15. Costache, A., Laine, K., Player, R.: Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In: Computer Security – ESORICS 2020. pp. 546–565 (2020)
16. Costache, A., Nürnberger, L., Player, R.: Optimisations and Tradeoffs for HElib. In: Topics in Cryptology – CT-RSA 2023. pp. 29–53 (2023)
17. Di Giusto, A., Marcolla, C.: Breaking the power-of-two barrier: noise estimation for bgv in ntt-friendly rings. Designs, Codes and Cryptography **93**(3), 467–502 (2025)
18. Ducas, L., Micciancio, D.: FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015. pp. 617–640. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
19. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. ePrint Archive (2012)
20. Feller, W.: An Introduction to Probability Theory and Its Applications. Vol. 1. Wiley (1950)
21. Gao, M., Zheng, H.: A critique on average-case noise analysis in rlwe-based homomorphic encryption. In: Proceedings of the 13th Workshop on Encrypted Computing & Applied Homomorphic Computing – WAHC 2025. pp. 26–37 (2025)
22. Gentry, C.: A fully homomorphic encryption scheme. Stanford university (2009)
23. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic Evaluation of the AES Circuit. In: Advances in Cryptology – CRYPTO 2012. pp. 850–867 (2012)
24. Halevi, S., Shoup, V.: Design and implementation of HElib: a homomorphic encryption library. ePrint Archive (2020)
25. Iliashenko, I.: Optimisations of fully homomorphic encryption. PhD thesis (2019)
26. Kim, A., Polyakov, Y., Zucca, V.: Revisiting homomorphic encryption schemes for finite fields. In: Advances in Cryptology–ASIACRYPT 2021. pp. 608–639 (2021)

27. Knuth, D.E.: The Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley, 3rd edn. (1997), section 4.6.3, Evaluation of Powers
28. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Advances in Cryptology – EUROCRYPT 2010. pp. 1–23 (2010)
29. Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F.H., Aaraj, N.: Survey on fully homomorphic encryption, theory, and applications. Proceedings of the IEEE **110**(10), 1572–1609 (2022)
30. Mono, J., Marcolla, C., Land, G., Güneysu, T., Aaraj, N.: Finding and evaluating parameters for BGV. In: International Conference on Cryptology in Africa. pp. 370–394 (2023)
31. Murphy, S., Player, R.: A central limit approach for ring-LWE noise analysis. IACR Communications in Cryptology **1**(2) (2024)
32. Okada, H., Player, R., Pohmann, S.: Homomorphic polynomial evaluation using galois structure and applications to BFV bootstrapping. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 69–100. Springer Nature Singapore, Singapore (2023)
33. Paterson, M.S., Stockmeyer, L.J.: On the number of nonscalar multiplications necessary to evaluate polynomials. SIAM Journal on Computing **2**(1), 60–66 (1973)
34. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6), 1–40 (2009)
35. Microsoft SEAL (release 3.4). https://github.com/Microsoft/SEAL (Oct 2019)
36. Shai Halevi and Victor Shoup: HElib. https://github.com/homenc/HElib
37. Stein, E.M., Shakarchi, R.: Fourier Analysis: An Introduction, Princeton Lectures in Analysis, vol. 1. Princeton University Press (2003)
38. Wang, Q., Wang, L.P.: A novel asymmetric bsgs polynomial evaluation algorithm under homomorphic encryption. In: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security. pp. 30–44 (2025)
39. Zhao, J., Zhang, J., Xiang, B., Wu, S., Deng, Y., Feng, D.: Faster polynomial evaluations for SIMD FHEs and application to BGV in HElib. Cryptology ePrint Archive, Paper 2026/1089 (2026), https://eprint.iacr.org/2026/1089

A Proof of Lemma 4

We show by contradiction that if π is not surjective, then $\dim_{\mathbb{F}_p} J \leq n/2$. Suppose $\pi(J) \subsetneq \mathbb{F}_p^2$, hence it is a subspace of dimension less than two over $\mathbb{F}_p$, which means that there exists $\rho, \beta \in \mathbb{F}_p$ such that $(\rho, \beta) \neq \{(0,0)\}$ and

$$\rho a|_{j_1} + \beta a|_{j_2} = 0 \quad \forall a \in J. \tag{11}$$

We now show that there exists a binomial b such that

$$ab = 0 \quad \forall a \in J. \tag{12}$$

Indeed, if we define $b \coloneqq \rho x^{-j_1} + \beta x^{-j_2} \in \mathcal{R}_p$, using the bilinearity of the product,

$$ab|_0 = \rho(ax^{-j_1})|_0 + \beta(ax^{-j_2})|_0 = \rho a|_{j_1} + \beta a|_{j_2} = 0, \tag{13}$$

where the last equality follows from Equation (11). Since J is an ideal, it is closed with respect to multiplication of any x^i, meaning that Equation (13) holds for

any coefficient i, which implies Equation (12). This means that $x^n + 1 \mid ab$ for all $a \in J$. Hence,

$$J \subseteq \mathrm{Ann}(b),$$

where $\mathrm{Ann}(b) \coloneqq \{a \in \mathcal{R}_p : ab = 0\}$ denotes the annihilator of b. Since $x^n = -1$ in $\mathcal{R}_p$, multiplication by any power of x is invertible. Therefore, multiplying b by a suitable power of x does not change the dimension of its annihilator. Let

$$r \coloneqq \min\{|j_1 - j_2|, n - |j_1 - j_2|\}.$$

Since $j_1 \neq j_2$, we have $1 \leq r \leq n/2$. Using $x^n = -1$, after multiplying b by a suitable power of x, we can write

$$b' = \rho' + \beta' x^r, \qquad \rho', \beta' \in \mathbb{F}_p^*.$$

Defining $g(x) = \gcd(x^n + 1, b'(x))$, it is possible to verify that

$$\dim_{\mathbb{F}_p} \mathrm{Ann}(b') = \deg(g).$$

Moreover, since g divides b', $\deg(g) \leq \deg(b') = r \leq \frac{n}{2}$, hence,

$$\dim_{\mathbb{F}_p} J \leq \dim_{\mathbb{F}_p} \mathrm{Ann}(b) = \dim_{\mathbb{F}_p} \mathrm{Ann}(b') \leq \frac{n}{2},$$

which contradicts the assumption $\dim_{\mathbb{F}_p} J > \frac{n}{2}$, concluding the proof.

B Proof of Theorem 6

Throughout the proof, for any ciphertext $\boldsymbol{c}$, we denote by $\nu_{\boldsymbol{c}}$ its critical quantity. According to Equation (8), we write $f(x)$ as

$$f(x) = \sum_{j=0}^{\rho} x^{jm} B_j(x), \text{ where } B_j(x) = a_{jm} + f_j(x) \text{ and } f_j(x) = \sum_{k=1}^{m-1} a_{jm+k} x^k,$$

with $a_l = 0$ for $l > \kappa$. We start by defining the multiplication levels at which powers of the ciphertexts involved in the evaluation are computed. For $1 \leq j \leq \rho$, we define $\ell_j := \lceil \log_2 m \rceil + \lceil \log_2 j \rceil$. Then,

- any baby-step power $\boldsymbol{c}^k$, with $1 \leq k \leq m$, lies at level $\lceil \log_2 k \rceil \leq \lceil \log_2 m \rceil$. In particular, $\boldsymbol{c}^m$ lies at level $\ell_1 = \lceil \log_2 m \rceil$;
- any giant-step power $(\boldsymbol{c}^m)^j = \boldsymbol{c}^{jm}$, with $2 \leq j \leq \rho$, lies at level ℓ_j, since $\boldsymbol{c}^m$ corresponds to level $\lceil \log_2 m \rceil$.

Each block $B_j(\boldsymbol{c})$ contains at most m monomials, and hence at most $m - 1$ additions. Specifically, the last block contains $\beta + 1$ monomials, while the other blocks contain at most m monomials. We start by distinguishing two cases: $\rho = 1$ and $\rho \geq 2$.

Case $\rho = 1$. Since $\rho = \lfloor \kappa/m \rfloor$, then $\kappa < 2m$. Moreover, $m = \lceil \sqrt{\kappa+1} \rceil$ gives $(m-1)^2 < \kappa + 1 < 2m + 1$, that is, $m < 4$. Since $\kappa \geq 2$, then $m \in \{2, 3\}$.
Now, we again distinguish between two cases: $\beta = 0$ and $\beta \neq 0$. Let us start by supposing that $\beta = 0$. In this case, we have $\kappa = m$. Since $m \in \{2, 3\}$, we must have $m = 2$: indeed, if $m = 3$, we would have $m = \lceil \sqrt{\kappa+1} \rceil = \lceil \sqrt{4} \rceil = 2$, a contradiction. Therefore, $\kappa = m = 2$ and

$$f(x) = a_1 x + a_2 x^2.$$

In particular, $f_1(x) = 0$ and $B_1(\boldsymbol{c}) = a_2$, so the block reduces to $a_2 \boldsymbol{c}^2$, which lies at level $\ell_1 = 1$. Since $\boldsymbol{c}^2$ is a square, Theorem 2 gives

$$\mathsf{Var}(\nu_{a_2 \boldsymbol{c}^2}|_i) \approx 4 n a_2^2 V_{\mathsf{ms}}^2.$$

Moreover, $B_0(\boldsymbol{c}) = a_1 \boldsymbol{c}$ is modulus-switched to level ℓ_1 before the addition, contributing V_{ms}. Thus,

$$\mathsf{Var}(\nu_f|_i) \leq V_{\mathsf{ms}} + 4 n a_2^2 V_{\mathsf{ms}}^2,$$

thus concluding this case. Instead, suppose now that $\beta \neq 0$, then $f_1(x) \neq 0$, since $a_\kappa \neq 0$ is the coefficient of x^β in $f_1(x)$. Thus:

- For B_1, the ciphertext $\boldsymbol{c}^m$ lies at level $\ell_1 = \lceil \log_2 m \rceil$, while $\boldsymbol{c}_{f_1}$ lies at level at most $\lceil \log_2(m-1) \rceil$. Since $m \in \{2, 3\}$, we have $\lceil \log_2(m-1) \rceil < \lceil \log_2 m \rceil$, so the modulus switching procedure is applied to $\boldsymbol{c}_{f_1}$ to obtain the same modulus as $\boldsymbol{c}^m$, and $\mathsf{Var}(\nu_{\boldsymbol{c}_{f_1}}|_i) = V_{\mathsf{ms}}$. By Theorem 3, $\mathsf{Var}(\nu_{\boldsymbol{c}^m \boldsymbol{c}_{f_1}}|_i) \approx 4 n V_{\mathsf{ms}}^2$, at level $\ell_1 + 1$. The summand $a_m \boldsymbol{c}^m$ lies at level ℓ_1 and is modulus-switched before the addition, hence $\mathsf{Var}(\nu_{\boldsymbol{c}^m B_1(\boldsymbol{c})}|_i) \approx V_{\mathsf{ms}} + 4 n V_{\mathsf{ms}}^2$.
- The block B_0 contains the baby-step powers $\boldsymbol{c}^k$, $k \leq m-1$, and hence lies at level at most $\lceil \log_2(m-1) \rceil < \ell_1 + 1$.

Thus, in order to sum the two blocks, the modulus switching procedure is applied to $B_0(\boldsymbol{c})$ and, by Proposition 3, $\mathsf{Var}(\nu_f|_i) \leq 2V_{\mathsf{ms}} + 4 n V_{\mathsf{ms}}^2$, concluding the case $\rho = 1$.

Case $\rho \geq 2$. We now analyze the blocks $B_j(\boldsymbol{c})$ separately, by considering the cases $j \in \{2, \ldots, \rho\}$, $j = 1$ and $j = 0$.

- $B_j(x)$, where $2 \leq j \leq \rho$. If $f_j(x) \neq 0$, all the ciphertext powers appearing in $f_j(x)$ lie at level at most $\lceil \log_2(m-1) \rceil \leq \lceil \log_2 m \rceil < \ell_j$, so that $\boldsymbol{c}_{f_j}$ lies at a level smaller than ℓ_j. Hence, in order to multiply $\boldsymbol{c}_{f_j}$ by $\boldsymbol{c}^{jm}$, modulus switch must be applied to $\boldsymbol{c}_{f_j}$ to obtain the same modulus as $\boldsymbol{c}^{jm}$. Thus $\mathsf{Var}(\nu_{\boldsymbol{c}_{f_j}}|_i) = V_{\mathsf{ms}}$. Now, to compute $\boldsymbol{c}^{jm} B_j(\boldsymbol{c})$, we apply Theorem 3 to $\boldsymbol{c}^{jm} \boldsymbol{c}_{f_j}$ and we obtain $\mathsf{Var}(\nu_{\boldsymbol{c}^{jm} \boldsymbol{c}_{f_j}}|_i) \approx 4 n V_{\mathsf{ms}}^2$ (which lies at level $\ell_j + 1$) and, since the remaining summand $a_{jm} \boldsymbol{c}^{jm}$ lies at level ℓ_j, the modulus-switch is applied before the addition. Hence the block $\boldsymbol{c}^{jm} B_j(\boldsymbol{c})$ satisfies

$$\mathsf{Var}(\nu_{\boldsymbol{c}^{jm} B_j(\boldsymbol{c})}|_i) \approx V_{\mathsf{ms}} + 4 n V_{\mathsf{ms}}^2, \qquad \text{at level } \ell_j + 1,$$

If, instead, $f_j(x) = 0$, the block reduces to the monomial $a_{jm}\boldsymbol{c}^{jm}$. Thus, by Theorems 2 and 3,

$$\mathsf{Var}(\nu_{\boldsymbol{c}^{jm}B_j(\boldsymbol{c})}|_i) \approx 4\,n\,a_{jm}^2 V_{\mathsf{ms}}^2, \qquad \text{at level } \ell_j,$$

Note that if $\beta \neq 0$, then $f_\rho(x) \neq 0$, and the maximum level is $d = \ell_\rho + 1$. Otherwise, if $\beta = 0$, then $f_\rho(x) = 0$, and $B_\rho \neq 0$, so $\boldsymbol{c}^{\rho m}B_\rho(\boldsymbol{c}) = a_\kappa \boldsymbol{c}^{\rho m}$ is nonzero and lies at level ℓ_ρ.

– For the block $B_1(x)$, $\boldsymbol{c}^m$ lies at level $\ell_1 = \lceil \log_2 m \rceil$. If $f_1(x) = 0$, the block reduces to $a_m\boldsymbol{c}^m$ and lies at level ℓ_1. If $f_1(x) \neq 0$ and $\lceil \log_2(m-1) \rceil < \lceil \log_2 m \rceil$, then $\boldsymbol{c}_{f_1}$ lies at a level smaller than ℓ_1 and the same computation as in the previous item applies. Otherwise $\boldsymbol{c}_{f_1}$ may already lie at level ℓ_1, and no modulus switching is applied before the product. In both cases $\boldsymbol{c}^m B_1(\boldsymbol{c})$ lies at level $\ell_1 + 1$.
Now, the maximal level satisfies $d \geq \ell_\rho$, with $d = \ell_\rho + 1$ if $\beta \geq 1$. Hence $\boldsymbol{c}^m B_1(\boldsymbol{c})$ lies at a level smaller than d whenever $\rho \geq 3$ (since $\ell_1 + 1 < \ell_\rho$) or $\rho = 2$ and $\beta \geq 1$ (because $\ell_1 + 1 = \ell_2 < d$). In these cases the modulus switching procedure is applied before the final addition and its contribution is V_{ms}.
In the remaining case, $\beta = 0$ and $\rho = 2$ (and $m = 3$). So, if $f_1(x) \neq 0$ the first case applies since $\ell_1 + 1 = \ell_2 = d$ and $\mathsf{Var}(\nu_{\boldsymbol{c}^m B_1(\boldsymbol{c})}|_i) \approx V_{\mathsf{ms}} + 4nV_{\mathsf{ms}}^2$. If $f_1(x) = 0$, the block lies at level $\ell_1 < d$ and contributes V_{ms}.
– The first block $B_0(x)$ requires no giant multiplication, and hence lies at a level smaller than d.

Finally, in order to sum all the terms of $f(\boldsymbol{c})$ together, all $\boldsymbol{c}^{jm}B_j(\boldsymbol{c})$ must be at the same level d. In particular, if $\beta \geq 1$,

– for any j such that $\lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil$ and $f_j(x) \neq 0$, the block $\boldsymbol{c}^{jm}B_j(\boldsymbol{c})$ already lies at level $d = \ell_\rho + 1$. Instead, if $f_j(x) = 0$, the block lies at level $\ell_\rho < d$ and is modulus-switched, contributing V_{ms}. Hence, let $\hat{\boldsymbol{c}} = \sum_{\lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil} \boldsymbol{c}^{jm}B_j(\boldsymbol{c})$. Thus, by Proposition 3,

$$\begin{aligned}\mathsf{Var}(\nu_{\hat{\boldsymbol{c}}}|_i) \;&\approx\; \#\{0 < j \leq \rho : \lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil\}\big(4nV_{\mathsf{ms}}^2 + V_{\mathsf{ms}}\big) \\ &= \#\{2^{\lceil \log_2 \rho \rceil - 1} < j \leq \rho\}\big(4nV_{\mathsf{ms}}^2 + V_{\mathsf{ms}}\big) \\ &= (\rho - 2^{\lceil \log_2 \rho \rceil - 1})\big(4nV_{\mathsf{ms}}^2 + V_{\mathsf{ms}}\big) = (\rho - A)\big(4nV_{\mathsf{ms}}^2 + V_{\mathsf{ms}}\big).\end{aligned}$$

– for any j such that $\lceil \log_2 j \rceil < \lceil \log_2 \rho \rceil$, and for $j = 0$, we apply the modulus switching technique to $\boldsymbol{c}^{jm}B_j(\boldsymbol{c})$. These blocks are (at most) $\rho + 1 - (\rho - A)$, since some blocks B_j may be zero.

Putting all together and applying Proposition 3, we obtain,

$$\mathsf{Var}(\nu_f|_i) \;\leq\; (1+A)V_{\mathsf{ms}} + (\rho - A)\big(4nV_{\mathsf{ms}}^2 + V_{\mathsf{ms}}\big) \;=\; (\rho+1)V_{\mathsf{ms}} + 4n(\rho - A)V_{\mathsf{ms}}^2.$$

If $\beta = 0$, note that $\ell_\rho = \ell_{\rho-1} + 1$, i.e. $\lceil \log_2 \rho \rceil > \lceil \log_2(\rho - 1) \rceil$, if and only if $\rho - 1$ is a power of 2. Note that if $f_j \neq 0$ for at least one $j \leq \rho - 1$ with $\lceil \log_2 j \rceil = \lceil \log_2(\rho - 1) \rceil$, then the maximal level is $d = \ell_{\rho-1} + 1$. Now,

– if $\rho - 1$ is not a power of 2, then $\ell_{\rho-1} = \ell_\rho$, and we have two cases:
  - if $f_j(x) \neq 0$, for at least a $j \leq \rho - 1$ such that $\lceil \log_2 j \rceil = \lceil \log_2(\rho-1) \rceil = \lceil \log_2 \rho \rceil$, then $d = \ell_\rho + 1$ and the block $\boldsymbol{c}^{\rho m} B_\rho(\boldsymbol{c})$ lies at level $\ell_\rho < d$ and is modulus-switched. The blocks lying at level d are those with $1 \leq \lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil$ and $j \leq \rho - 1$, which are $\rho - 1 - 2^{\lceil \log_2 \rho \rceil - 1} = \rho - A - 1$. Since the total number of blocks is (at most) $\rho + 1$, we have

$$\mathsf{Var}(\nu_f|_i) \leq (\rho+1)V_{\mathsf{ms}} + 4n\big(\rho - 1 - 2^{\lceil \log_2 \rho \rceil - 1}\big)V_{\mathsf{ms}}^2.$$

  - if $f_j(x) = 0$ for every $j \leq \rho - 1$ such that $\lceil \log_2 j \rceil = \lceil \log_2(\rho-1) \rceil = \lceil \log_2 \rho \rceil$, then $d = \ell_\rho$. The blocks $\boldsymbol{c}^{jm} B_j(\boldsymbol{c})$ lying at level d are the ones with
    * $A < j \leq \rho$, which are reduced to $a_{jm}\boldsymbol{c}^{jm}$ and each of them contributes $4n\, a_{jm}^2 V_{\mathsf{ms}}^2$;
    * $\lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil - 1$, i.e. $2^{\lceil \log_2 \rho \rceil - 2} < j \leq 2^{\lceil \log_2 \rho \rceil - 1}$, which are at most $A/2 = 2^{\lceil \log_2 \rho \rceil - 2}$ and their contribution is $V_{\mathsf{ms}} + 4nV_{\mathsf{ms}}^2$ each.

    All the remaining blocks, which are at most $\rho - A/2 - (\rho - A - 1) = 1 + A/2$, are modulus-switched, thus

$$\mathsf{Var}(\nu_f|_i) \leq \big(1 + A\big)V_{\mathsf{ms}} + n\Big(2A + \sum_{A < j \leq \rho} 4\, a_{jm}^2\Big)V_{\mathsf{ms}}^2,$$

– if $\rho - 1 \geq 2$ is a power of 2, then no $j < \rho$ satisfies $\lceil \log_2 j \rceil = \lceil \log_2 \rho \rceil$, hence $d = \ell_\rho$. The blocks lying at level d are $a_\kappa \boldsymbol{c}^{\rho m}$ and those with $\frac{\rho-1}{2} < j \leq \rho - 1$ and $f_j(x) \neq 0$, which are at most $\frac{\rho-1}{2}$. The blocks in this range with $f_j(x) = 0$ lie at level $\ell_j < d$ and are modulus-switched. Hence,

$$\mathsf{Var}(\nu_f|_i) \leq \rho\, V_{\mathsf{ms}} + 2n(\rho-1)V_{\mathsf{ms}}^2 + 4n\, a_\kappa^2 V_{\mathsf{ms}}^2.$$

– if $\rho = 2$, then thanks to the contribution of B_0 and B_1, we have

$$\mathsf{Var}(\nu_f|_i) \leq 2V_{\mathsf{ms}} + 4nV_{\mathsf{ms}}^2 + 4n\, a_\kappa^2 V_{\mathsf{ms}}^2.$$