Supplementary information for

Physics-based compact model for 2D TMD FETs with full-range validation from single device to circuit

Dokyoung Lee, Jeongyun Jang, Jimin Han, Sae Rim Kim, Jiwon Baik, Hayoung Roh, and Sungho Kim

Division of Electronic & Semiconductor Engineering, Ewha Womans University, Seoul 03670, Republic of Korea

Corresponding author: *Sungho Kim*, email: sunghok@ewha.ac.kr

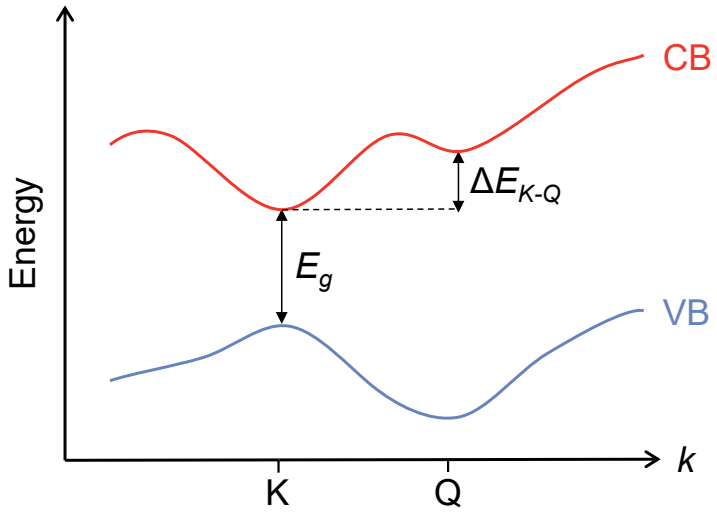

Figure S1. The conduction band (CB) and valence band (VB) structures of a typical TMD material. In the conduction band, two prominent minima are observed: one at the K point and another at the Q point of the Brillouin zone. The energy separation between these valleys, denoted as ΔE_{K-Q}, is relatively small—typically on the order of a few tens of meV—depending on the specific TMD material and its thickness.

Due to the small ΔE_{K-Q}, the thermal population of electrons in the Q valley becomes non-negligible even at room temperature. As a result, both valleys contribute significantly to the total carrier concentration in the conduction band. Neglecting the contribution of the Q valley can lead to substantial underestimation of the mobile electron density, particularly under high gate bias or elevated temperature conditions where the occupation of higher-energy valleys is enhanced.

Therefore, for accurate modeling of carrier statistics in 2D TMD-based FETs, it is essential to incorporate the density-of-states contributions from both the K and Q valleys. This multivalley formulation described in Equation (6) enables more precise evaluation of the surface potential and drain current.

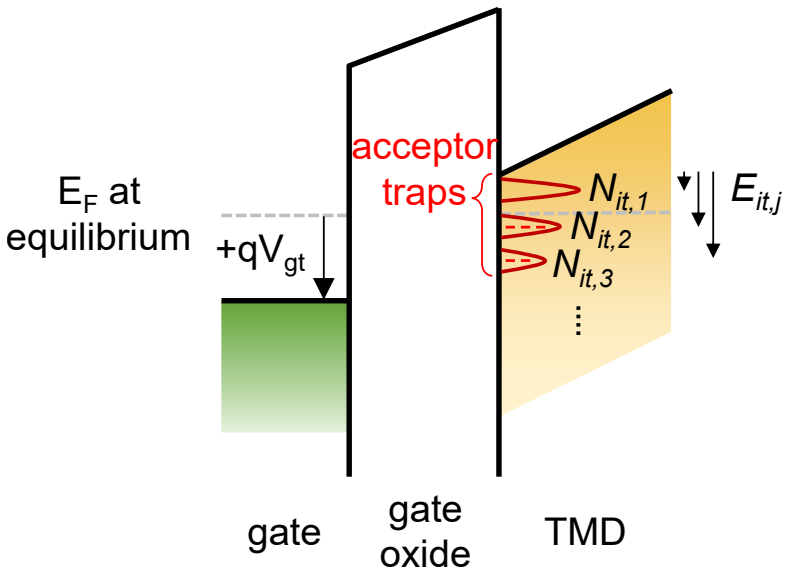

Figure S2. At the interface between the 2D TMD channel and the gate oxide, defect states—primarily originating from structural imperfections such as sulfur vacancies in MoS_2 or oxygen vacancies in high-k dielectrics—give rise to acceptor-like trap states that reside energetically within the bandgap. In our model, the total density of occupied interface trap states N_{it} is approximated as a summation over multiple discrete energy levels, each modeled as a delta-function–like localized state. This discretized multi-level formulation serves as a computationally efficient alternative to continuous trap distributions while still enabling accurate modeling of the electrostatic perturbations induced by interface traps.

changed parameters	value	fitting parameter	Value
ε_{tox} (HfO_2)	22	N_{imp}	5.98×10^{11} cm^{-2}
t_{box}	285 nm	D_{it1}	7.27×10^{11} cm^{-2}
t_{tox}	20 nm	D_{it2}	8.11×10^{10} cm^{-2}
L	1 μm	D_{it3}	2.03×10^{11} cm^{-2}
W	1 μm	E_{it1}	0.019 eV
$\Delta\Psi_t$ (Al−MoS_2)	-0.24 (D mode)	E_{it2}	0.275 eV
		E_{it3}	1.105 eV
$\Delta\Psi_t$ (Pd−MoS_2)	0.52 (E mode)	μ_0	9.52 $cm^2V^{-1}s^{-1}$
		θ_1	2.762
		θ_2	0.0227

Table S1. The simulation parameters that differ from those listed in Table 1, specifically for reproducing the measured circuit characteristics reported in Ref. [28]. While the majority of model parameters used in our device remain applicable, certain parameters—particularly those related to device geometry, gate stack materials, and contact properties—were adjusted to reflect the physical configurations of the MoS_2 FETs used in the referenced experimental circuits.

	interface trap effect	quantum capacitance effect	SPICE compatibility (data obtained by SPICE)
Ref. 11	X	X	X
Ref. 12	O	O	△
Ref. 13	O	O	△
Ref. 14	O	X	X
Ref. 15	O	X	X
Ref. 16	O	X	△
Ref. 17	O	X	X
Ref. 18	O	O	X
Ref. 19	O	O	X
Ref. 20	O	O	X
This work	**O**	**X**	**O**

Table S2. Comparison of representative compact models for 2D TMD FETs in terms of interface-trap effect, quantum-capacitance effect, and SPICE compatibility. Most previous models include interface-trap effects, whereas only a few recent studies explicitly account for quantum capacitance. Although several works claimed SPICE compatibility, their simulation data were generally not obtained directly from SPICE. In contrast, the present model achieves complete SPICE compatibility, with all simulation results—ranging from single-device to circuit-level behavior—generated within the SPICE environment.

Supplementary Note 1. Closed-form formulation of surface potential using the Lambert W function

To obtain an explicit analytical expression for the surface potential ϕ, we begin with the substitution of Equation (4) into Equation (7):

$$\frac{C_T}{q}(\alpha V_{geff} - \phi) = N_{DOS} \exp\left(\frac{\phi - V}{v_{th}}\right) \tag{S1}$$

Rewriting the above expression:

$$\alpha V_{geff} - \phi = \frac{qN_{DOS}}{C_T} \exp\left(\frac{\phi - V}{v_{th}}\right) \tag{S2}$$

We now define a new variable x:

$$x \equiv \alpha V_{geff} - \phi \quad \Rightarrow \quad \phi = \alpha V_{geff} - x \tag{S3}$$

Substituting into the exponential term:

$$x = \frac{qN_{DOS}}{C_T} \exp\left(\frac{\alpha V_{geff} - x - V}{v_{th}}\right) \tag{S4}$$

Rewriting the expression:

$$\frac{x}{v_{th}} \exp\left(\frac{x}{v_{th}}\right) = \frac{qN_{DOS}}{C_T v_{th}} \exp\left(\frac{\alpha V_{geff} - V}{v_{th}}\right) \tag{S5}$$

Accorting to the definition of the Lambert W function:

$$\frac{x}{v_{th}} = W\left[\frac{qN_{DOS}}{C_T v_{th}} \exp\left(\frac{\alpha V_{geff} - V}{v_{th}}\right)\right] \tag{S6}$$

Finally, substituting back x, we obtain the explicit expression for the surface potential:

$$\phi = \alpha V_{geff} - v_{th} \cdot W\left[\frac{qN_{DOS}}{C_T v_{th}} \exp\left(\frac{\alpha V_{geff} - V}{v_{th}}\right)\right] \tag{S7}$$

Supplementary Note 2. Parameter extraction via particle swarm optimization (PSO)

Accurate simulation of 2D FETs requires the determination of physically meaningful parameters in the compact model, including impurity density (N_{imp}), interface trap densities (D_{it}), trap energy levels (E_{it}), low-field mobility (μ_0), and field-dependent mobility coefficients (θ_1, θ_2). Although these parameters can, in principle, be extracted through direct experimental techniques such as capacitance–voltage (C–V), current–voltage (I–V), deep-level transient spectroscopy (DLTS), or spectroscopic methods like photoluminescence (PL), scanning transmission electron microscopy (STEM), and X-ray photoelectron spectroscopy (XPS), such approaches require complex instrumentation or dedicated sample preparation. As an alternative, we extract these parameters by solving a combinatorial optimization problem that identifies the parameter set yielding the best fit to the experimentally measured device characteristics.

Due to the highly nonlinear and nonconvex nature of the compact model equations, conventional gradient-based optimization methods are prone to becoming trapped in local minima or may fail to converge. To overcome this limitation, we employed the Particle Swarm Optimization (PSO) algorithm, a population-based metaheuristic that efficiently explores high-dimensional parameter spaces to locate the global minimum of the cost function. In the PSO algorithm, a population (swarm) of candidate solutions (particles) is randomly initialized within predefined parameter bounds. Each particle represents a specific set of compact model parameters:

$$p = [N_{imp}, D_{it,j}, E_{it,j}, \mu_0, \theta_1, \theta_2]$$

The fitting process is formulated as a minimization of the following cost function $J(p)$, which computes the sum of squared logarithmic errors between the measured drain current ($I_{DS,exp}$) and the simulated drain current from the compact model ($I_{DS,model}$):

$$J(p) = \sum_i \left[\log\left(I_{DS,model}(V_{gt,i}, p) + \varepsilon\right) - \log\left(I_{DS,exp}(V_{gt,i}) + \varepsilon\right) \right]^2$$

where $\varepsilon = 10^{-11}$ is added to prevent numerical instability when computing logarithms of very small current values.

During optimization, each particle iteratively updates its position in the parameter space based on its own best-known position and the globally best-known position of the swarm. The velocity and position updates follow standard PSO update rules. In our implementation, a swarm size of 200 particles and a maximum of 2000 iterations were employed to ensure sufficient coverage of the search space. The parameter boundaries were set based on physical plausibility and prior empirical knowledge:

$$N_{imp} \in \left[10^{11}, 5\times10^{12}\right] cm^{-2},\ D_{it,j} \in \left[10^{10}, 10^{12}\right] cm^{-2},\ E_{it,j} \in \left[0, 50v_{th}\right] eV,$$

$$\mu_0 \in \left[1, 50\right] cm^2 / V \cdot s,\ \theta_{1,2} \in \left[-1, 1\right]$$

The PSO algorithm was implemented using MATLAB's Global Optimization Toolbox. Through this process, the optimal fitting parameter set shown in Table 1 and Table S1 that minimizes the cost function is efficiently extracted.

The following pseudocode outlines the parameter-extraction procedure used to fit the 2D-FET compact model to experimental transfer characteristic using the Particle Swarm Optimization (PSO) algorithm. This representation is simulator-independent and can be implemented in MATLAB, Python, or any equivalent optimization framework.

```
Load experimental data:
    Read 'data.xlsx' file.
    Vgt_exp ← first column  (experimental gate voltage)
    Ids_exp ← second column (measured drain current)

Define cost function:
    epsilon = 1×10⁻¹¹  // offset to avoid log(0)
    For a given parameter vector p:
        Simulate drain current using the compact model:
            Ids_model = compactModelIds(Vgt_exp, p)
        Compute error:
            cost(p) = Σ [ log(Ids_model + epsilon) – log(Ids_exp + epsilon) ]²

Define fitting parameters:
    Parameter order:
```

```
        [Nimp, Dit1, Dit2, Dit3, Eit1, Eit2, Eit3, μ0, θ1, θ2]
    Set parameter dimension:
        nParams = 10

Specify physical boundary conditions:
    Lower bounds (lb) = [1e15, 1e14, 1e14, 1e14, 0, 0, 0, 1e−4, −1, −1]
    Upper bounds (ub) = [5e16, 1e16, 1e16, 1e16, 50·vth, 50·vth, 50·vth, 1e−3, 1, 1]

Configure PSO settings:
    swarmSize = 200
    maxIterations = 2000
    enableParallel = true
    displayMode = "iterative"

Execute PSO optimization:
    Initialize swarm of particles within [lb, ub]
    For each iteration up to maxIterations:
        Evaluate cost(particle_i) for all particles
        Update particle velocity and position according to PSO rules:
            v_i(t+1) = w·v_i(t) + c₁·r₁·(p_best_i − p_i) + c₂·r₂·(g_best − p_i)
            p_i(t+1) = p_i(t) + v_i(t+1)
        Update personal best (p_best_i) and global best (g_best)
    After convergence:
        p_opt ← g_best parameters
        fval ← minimum cost value

Output:
    Print optimized parameters (p_opt)
    Print final cost (fval)
```

Supplementary Note 3. Pseudocode of the SPICE-compatible 2D FET compact model

To facilitate model implementation across different SPICE environments, the computational flow of the proposed compact model is summarized below in pseudocode form. The pseudocode describes the essential parameter definitions, intermediate variables, and the sequence of calculations, while avoiding simulator-specific syntax.

```
Define physical constants:
    q = 1.6×10⁻¹⁹  (elementary charge)
    vth = kT/q ≈ 0.026 V  (thermal voltage at 300 K)
    ε₀ = 8.85×10⁻¹² F/m

Define geometry and material parameters:
    W, L = channel width and length
    TTOX, TBOX = top and bottom oxide thickness
    εTOX, εBOX = dielectric constants of top and bottom oxides
    Δψ_t, Δψ_b = flat-band voltage offsets
    NDOS = 2D density of states
    Nimp = impurity charge density
    Dit1, Dit2, Dit3 = interface trap densities
    Eit1, Eit2, Eit3 = trap energy levels relative to Ec
    μ0 = low-field mobility
    θ1, θ2 = field-dependent mobility coefficients

Compute capacitances and electrostatic parameters:
    Ctox = εTOX / TTOX
    Cbox = εBOX / TBOX
    CT = Ctox + Cbox
    α = Ctox / CT

Calculate effective gate voltage:
    Vgeff = V(Gt) + [(1–α)/α]·V(Gb) – [Δψ_t + ((1–α)/α)·Δψ_b – (q·Nimp)/(α·CT)]

Evaluate surface potentials:
    For source and drain terminals (indexed by terminal voltage Vt):
        Temp = (q·NDOS)/(CT·vth) · exp[(α·Vgeff – Vt)/vth]
        ϕ = α·Vgeff – vth · W_approx(Temp)
    where W_approx(x) is the analytical approximation of the Lambert W function,
    defined by a Taylor-series expansion for x < 0.1 and
    a nested-logarithmic expression for x ≥ 0.1.

Compute mid-channel potential for trap evaluation:
    Temp_Nit = (q·NDOS)/(CT·vth) · exp[(α·Vgeff – V(D)/2)/vth]
    ϕ_Nit = α·Vgeff – vth · W_approx(Temp_Nit)

Calculate interface trap charge density:
    Nit = Σ over j = 1→3 [ Dit_j / (1 + exp(–Eit_j/vth) · exp(–(ϕ_Nit – V(D)/2)/vth)) ]

Determine effective mobility including vertical-field degradation:
    μ_eff = μ0 / [1 + θ1·Vgeff + θ2·(Vgeff)²]

Compute drain current:
    IDS = (W/L) · μ_eff · CT · [ (α·Vgeff + vth – q·Nit/CT) · (ϕ_D – ϕ_S) – 0.5·(ϕ_D² – ϕ_S²) ]

Output:
```

Drain–source current = IDS

Auxiliary functions:

LambertAprox(x) = ln((50/47)·x / ln((60/47)·x / ln((120/47)·x / ln(1 + (120/47)·x))))

LambertTaylor(x) = x – x² + (3/2)·x³