Online Supplement Part I: Derivation of model-implied outcome variance for each type of multilevel regression mixture model with random effects (L1MIX, L2MIX, and L1L2MIX)

In Part I of the Online Supplement, we derive the model-implied outcome variance for each type of multilevel regression mixture model with random effects, including specifications with classes at only level-1 (L1MIX in manuscript Equation (2)), classes at only level-2 (L2MIX in manuscript Equation (3)), and classes at both level-1 and level-2 (L1L2MIX in manuscript Equation (5)). The (print) Appendices A1-A7 show how to use our decomposition of variance to form each of the R-squared measures in the R-squared framework, as described in the manuscript. Here we begin by deriving the model-implied outcome variance for the model with classes at both level-1 and level-2 (L1L2MIX), and show how this most general expression reduces for the models with classes at only one level. Here we assume level-1 predictors are group mean centered for reasons described in the manuscript. Note that all terms in the following derivations are defined either below or in the (print) Appendices A1 and A4 which each contain a list of notation and symbol definitions.

Derivation of the total model-implied outcome variance for the multilevel regression mixture model with classes at both levels (L1L2MIX). For the L1L2MIX (manuscript Equation (5)), we first derive the total model-implied outcome variance:

$$\text{var}(y_{ij|c_{ij}=k,d_j=h}) = \text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} + \mathbf{x}_j^{b\prime}\boldsymbol{\gamma}^{b(kh)} + \mathbf{w}_{ij}'\mathbf{u}_j + \varepsilon_{ij}) \quad \text{(S1)}$$

In this expression, the following pairs of terms are uncorrelated: the residuals (both level-1 and level-2) and predictors, the cluster-mean-centered level-1 predictors (with no across-cluster variance) and level-2 predictors, and the level-1 residuals and level-2 residuals. Hence, we can rewrite Equation (S1) as

$$\text{var}(y_{ij|c_{ij}=k,d_j=h}) = \underbrace{\text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)})}_{(a)} + \underbrace{\text{var}(\mathbf{x}_j^{b\prime}\boldsymbol{\gamma}^{b(kh)})}_{(b)} + \underbrace{\text{var}(\mathbf{w}_{ij}'\mathbf{u}_j)}_{(c)} + \underbrace{\text{var}(\varepsilon_{ij})}_{(d)} \quad \text{(S2)}$$

We will next solve for parts (a)-(d) in turn.

Using the law of total variance and conditioning on class-combination membership, we can express part (a) as:

$$\begin{aligned}
\text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)}) &= E[\text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} \mid c_{ij}=k, d_j=h)] + \text{var}(E[\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} \mid c_{ij}=k, d_j=h]) \\
&= E[\boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(kh)}] + \text{var}(E[\mathbf{x}_{ij}^{w\prime}]\boldsymbol{\gamma}^{w(kh)}) \\
&= E[\boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(kh)}] + \text{var}(0) \\
&= E[\boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(kh)}] \\
&= E[tr(\boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(kh)})] \\
&= E[tr(\boldsymbol{\gamma}^{w(kh)}\boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^w)] \\
&= tr(E[\boldsymbol{\gamma}^{w(kh)}\boldsymbol{\gamma}^{w(kh)\prime}]\boldsymbol{\Phi}^w)
\end{aligned}$$

$$\begin{aligned}
&= tr((\text{var}(\boldsymbol{\gamma}^{w(kh)}) + E[\boldsymbol{\gamma}^{w(kh)}]E[\boldsymbol{\gamma}^{w(kh)\prime}])\boldsymbol{\Phi}^{w}) \\
&= tr((\boldsymbol{\Psi}^{w} + \boldsymbol{\gamma}^{w(\bullet\bullet)}\boldsymbol{\gamma}^{w(\bullet\bullet)\prime})\boldsymbol{\Phi}^{w}) \\
&= tr(\boldsymbol{\Psi}^{w}\boldsymbol{\Phi}^{w} + \boldsymbol{\gamma}^{w(\bullet\bullet)}\boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}) \\
&= tr(\boldsymbol{\Psi}^{w}\boldsymbol{\Phi}^{w}) + tr(\boldsymbol{\gamma}^{w(\bullet\bullet)}\boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}) \\
&= tr(\boldsymbol{\Psi}^{w}\boldsymbol{\Phi}^{w}) + tr(\boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)}) \\
&= tr(\boldsymbol{\Psi}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)}
\end{aligned} \tag{S3}$$

(see definitions of individual terms provided in Appendix Table A4). Note that $\boldsymbol{\Psi}^{w}$ represents the across-class-combination covariance matrix of class-combination-specific regression coefficients, computed as $\text{var}(\boldsymbol{\gamma}^{w(kh)}) = \sum_{h=1}^{H}\sum_{k=1}^{K}\pi^{kh}\boldsymbol{\gamma}^{w(kh)}\boldsymbol{\gamma}^{w(kh)\prime} - \left(\sum_{h=1}^{H}\sum_{k=1}^{K}\pi^{kh}\boldsymbol{\gamma}^{w(kh)}\right)\left(\sum_{h=1}^{H}\sum_{k=1}^{K}\pi^{kh}\boldsymbol{\gamma}^{w(kh)\prime}\right)$. We can partition this further by replacing $\boldsymbol{\Psi}^{w}$ with the sum of its orthogonal components $\boldsymbol{\Omega}^{w}$ (coefficient covariation across level-1 class within level-2 class) and $\mathbf{Z}^{w}$ (coefficient covariation across level-2 class).

$$\begin{aligned}
\text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)}) &= tr(\boldsymbol{\Psi}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)} \\
&= tr((\boldsymbol{\Omega}^{w} + \mathbf{Z}^{w})\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)} \\
&= tr(\boldsymbol{\Omega}^{w}\boldsymbol{\Phi}^{w} + \mathbf{Z}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)} \\
&= tr(\boldsymbol{\Omega}^{w}\boldsymbol{\Phi}^{w}) + tr(\mathbf{Z}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\bullet\bullet)}
\end{aligned} \tag{S4}$$

We next express part (b) using steps similar to those for part (a)

$$\begin{aligned}
\text{var}(\mathbf{x}_{ij}^{b\prime}\boldsymbol{\gamma}^{b(kh)}) &= E[\text{var}(\mathbf{x}_{ij}^{b\prime}\boldsymbol{\gamma}^{b(kh)} \mid c_{ij}=k, d_j=h)] + \text{var}(E[\mathbf{x}_{ij}^{b\prime}\boldsymbol{\gamma}^{b(kh)} \mid c_{ij}=k, d_j=h]) \\
&= E[\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)}] + \text{var}(E[\mathbf{x}_{ij}^{b\prime}]\boldsymbol{\gamma}^{b(kh)}) \\
&= E[\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)}] + \text{var}(\mathbf{m}^{b\prime}\boldsymbol{\gamma}^{b(kh)}) \\
&= E[\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)}] + \mathbf{m}^{b\prime}\,\text{var}(\boldsymbol{\gamma}^{b(kh)})\mathbf{m}^{b} \\
&= E[\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)}] + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b} \\
&= E[tr(\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)})] + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b} \\
&= E[tr(\boldsymbol{\gamma}^{b(kh)}\boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b})] + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b} \\
&= tr(E[\boldsymbol{\gamma}^{b(kh)}\boldsymbol{\gamma}^{b(kh)\prime}]\boldsymbol{\Phi}^{b}) + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b} \\
&= tr((\text{var}(\boldsymbol{\gamma}^{b(kh)}) + E[\boldsymbol{\gamma}^{b(kh)}]E[\boldsymbol{\gamma}^{b(kh)\prime}])\boldsymbol{\Phi}^{b}) + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b} \\
&= tr((\boldsymbol{\Psi}^{b} + \boldsymbol{\gamma}^{b(\bullet\bullet)}\boldsymbol{\gamma}^{b(\bullet\bullet)\prime})\boldsymbol{\Phi}^{b}) + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^{b}\mathbf{m}^{b}
\end{aligned}$$

$$
\begin{aligned}
&= tr(\boldsymbol{\Psi}^b\boldsymbol{\Phi}^b + \boldsymbol{\gamma}^{b(\cdot\cdot)}\boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b) + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^b\mathbf{m}^b \\
&= tr(\boldsymbol{\Psi}^b\boldsymbol{\Phi}^b + \boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\cdot\cdot)}) + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^b\mathbf{m}^b \\
&= tr(\boldsymbol{\Psi}^b\boldsymbol{\Phi}^b) + \boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\cdot\cdot)} + \mathbf{m}^{b\prime}\boldsymbol{\Psi}^b\mathbf{m}^b \\
&= tr((\boldsymbol{\Omega}^b + \mathbf{Z}^b)\boldsymbol{\Phi}^b) + \boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\cdot\cdot)} + \mathbf{m}^{b\prime}(\boldsymbol{\Omega}^b + \mathbf{Z}^b)\mathbf{m}^b \\
&= tr(\boldsymbol{\Omega}^b\boldsymbol{\Phi}^b + \mathbf{Z}^b\boldsymbol{\Phi}^b) + \boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\cdot\cdot)} + \mathbf{m}^{b\prime}\boldsymbol{\Omega}^b\mathbf{m}^b + \mathbf{m}^{b\prime}\mathbf{Z}^b\mathbf{m}^b \\
&= tr(\boldsymbol{\Omega}^b\boldsymbol{\Phi}^b) + tr(\mathbf{Z}^b\boldsymbol{\Phi}^b) + \boldsymbol{\gamma}^{b(\cdot\cdot)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\cdot\cdot)} + \mathbf{m}^{b\prime}\boldsymbol{\Omega}^b\mathbf{m}^b + \mathbf{m}^{b\prime}\mathbf{Z}^b\mathbf{m}^b
\end{aligned} \tag{S5}
$$

where $\text{var}(\boldsymbol{\gamma}^{b(kh)}) = \boldsymbol{\Psi}^b = \boldsymbol{\Omega}^b + \mathbf{Z}^b$. We can express part (c) using the law of total variance, conditioning on both random effects and class-combination, as such

$$
\begin{aligned}
\text{var}(\mathbf{w}_{ij}'\mathbf{u}_j) &= E[\text{var}(\mathbf{w}_{ij}'\mathbf{u}_j \mid \mathbf{u}_j, c_{ij}=k, d_j=h)] + E[\text{var}(E[\mathbf{w}_{ij}'\mathbf{u}_j \mid \mathbf{u}_j, c_{ij}=k, d_j=h] \mid \mathbf{u}_j)] + \text{var}(E[\mathbf{w}_{ij}'\mathbf{u}_j \mid \mathbf{u}_j]) \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + E[\text{var}(E[\mathbf{w}_{ij}']\mathbf{u}_j \mid \mathbf{u}_j)] + \text{var}(E[\mathbf{w}_{ij}']\mathbf{u}_j) \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + 0 + \text{var}(E[\mathbf{w}_{ij}']\mathbf{u}_j) \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \text{var}(\mathbf{m}^{r\prime}\mathbf{u}_j) \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \mathbf{m}^{r\prime}\,\text{var}(\mathbf{u}_j)\mathbf{m}^r \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \mathbf{m}^{r\prime}(E[\text{var}(\mathbf{u}_j \mid c_{ij}=k, d_j=h)] + \text{var}(E[\mathbf{u}_j \mid c_{ij}=k, d_j=h]))\mathbf{m}^r \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \mathbf{m}^{r\prime}(E[\mathbf{T}^h] + \text{var}(0))\mathbf{m}^r \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \mathbf{m}^{r\prime}\mathbf{T}^{\cdot}\mathbf{m}^r \\
&= E[\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j] + \tau_{00}^{\cdot} \\
&= E[tr(\mathbf{u}_j'\boldsymbol{\Phi}^r\mathbf{u}_j)] + \tau_{00}^{\cdot} \\
&= E[tr(\mathbf{u}_j\mathbf{u}_j'\boldsymbol{\Phi}^r)] + \tau_{00}^{\cdot} \\
&= tr(E[\mathbf{u}_j\mathbf{u}_j']\boldsymbol{\Phi}^r) + \tau_{00}^{\cdot} \\
&= tr(E[E[\mathbf{u}_j\mathbf{u}_j' \mid c_{ij}=k, d_j=h]]\boldsymbol{\Phi}^r) + \tau_{00}^{\cdot} \\
&= tr(E[\mathbf{T}^h]\boldsymbol{\Phi}^r) + \tau_{00}^{\cdot} \\
&= tr(\mathbf{T}^{\cdot}\boldsymbol{\Phi}^r) + \tau_{00}^{\cdot}
\end{aligned} \tag{S6}
$$

Lastly, we can express part (d) using the law of total variance and conditioning on class-combination

$$
\begin{aligned}
\text{var}(\varepsilon_{ij}) &= E[\text{var}(\varepsilon_{ij} \mid c_{ij}=k, d_j=h)] + \text{var}(E[\varepsilon_{ij} \mid c_{ij}=k, d_j=h]) \\
&= E[\theta^{kh}] + \text{var}(0) \\
&= \theta^{\cdot\cdot}
\end{aligned} \tag{S7}
$$

Thus,

$$\begin{aligned}\text{var}(y_{ij|c_{ij}=k,d_j=h}) &= \text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} + \mathbf{x}_j^{b\prime}\boldsymbol{\gamma}^{b(kh)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_j + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(\bullet\bullet)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(\bullet\bullet)} + tr(\boldsymbol{\Omega}^w\boldsymbol{\Phi}^w) + tr(\mathbf{Z}^w\boldsymbol{\Phi}^w) + \boldsymbol{\gamma}^{b(\bullet\bullet)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\bullet\bullet)} \\ &\quad + tr(\boldsymbol{\Omega}^b\boldsymbol{\Phi}^b) + tr(\mathbf{Z}^b\boldsymbol{\Phi}^b) + \mathbf{m}^{b\prime}\boldsymbol{\Omega}^b\mathbf{m}^b + \mathbf{m}^{b\prime}\mathbf{Z}^b\mathbf{m}^b + tr(\mathbf{T}^{\bullet}\boldsymbol{\Phi}^r) + \tau_{00}^{\bullet} + \theta^{\bullet\bullet}\end{aligned} \qquad \text{(S8)}$$

As shown in (print) Appendix Table A5, this model-implied total outcome variance in Equation (S8) serves as the denominator for all total R-squared measures for the L1L2MIX model. All other R-squared measure denominators (for each type of measure and for each type of model) are special cases of this most general expression, as we show next.

Derivation of the level-2-class-specific model-implied outcome variance for the multilevel regression mixture model with classes at both levels (L1L2MIX). The denominator for the level-2-class-specific R-squared measures (i.e., the model-implied within-level-2-class-*h* outcome variance) for the L1L2MIX is derived as follows:

$$\begin{aligned}\text{var}_{k|h}(y_{ij|c_{ij}=k,d_j=h}) &= \text{var}_{k|h}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} + \mathbf{x}_j^{b\prime}\boldsymbol{\gamma}^{b(kh)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_j + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(\bullet h)\prime}\boldsymbol{\Phi}^w\boldsymbol{\gamma}^{w(\bullet h)} + tr(\boldsymbol{\Omega}^{w(h)}\boldsymbol{\Phi}^w) + \boldsymbol{\gamma}^{b(\bullet h)\prime}\boldsymbol{\Phi}^b\boldsymbol{\gamma}^{b(\bullet h)} \\ &\quad + tr(\boldsymbol{\Omega}^{b(h)}\boldsymbol{\Phi}^b) + \mathbf{m}^{b\prime}\boldsymbol{\Omega}^{b(h)}\mathbf{m}^b + tr(\mathbf{T}^h\boldsymbol{\Phi}^r) + \tau_{00}^h + \theta^{\bullet h}\end{aligned} \qquad \text{(S9)}$$

where

$$\boldsymbol{\gamma}^{w(\bullet h)} = \sum_{k=1}^{K}\pi^{kh}\boldsymbol{\gamma}^{w(kh)} \qquad \text{(S10)}$$

$$\boldsymbol{\Omega}^{w(h)} = \sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{w(kh)} - \boldsymbol{\gamma}^{w(\bullet h)})(\boldsymbol{\gamma}^{w(kh)} - \boldsymbol{\gamma}^{w(\bullet h)})' - \left(\sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{w(kh)} - \boldsymbol{\gamma}^{w(\bullet h)})\right)\left(\sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{w(kh)} - \boldsymbol{\gamma}^{w(\bullet h)})'\right) \qquad \text{(S11)}$$

$$\boldsymbol{\gamma}^{b(\bullet h)} = \sum_{k=1}^{K}\pi^{kh}\boldsymbol{\gamma}^{b(kh)} \qquad \text{(S12)}$$

$$\boldsymbol{\Omega}^{b(h)} = \sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{b(kh)} - \boldsymbol{\gamma}^{b(\bullet h)})(\boldsymbol{\gamma}^{b(kh)} - \boldsymbol{\gamma}^{b(\bullet h)})' - \left(\sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{b(kh)} - \boldsymbol{\gamma}^{b(\bullet h)})\right)\left(\sum_{k=1}^{K}\pi^{kh}(\boldsymbol{\gamma}^{b(kh)} - \boldsymbol{\gamma}^{b(\bullet h)})'\right) \qquad \text{(S13)}$$

$$\theta^{\bullet h} = \sum_{k=1}^{K}\pi^{kh}\theta^{kh} \qquad \text{(S14)}$$

Equation (S9) differs from the expression in Equation (S8) in that *h* is held constant and thus there is no variance attributable to v_1^h, v_2^h, or m^h, nor is there marginalization across all *h*. As shown in (print) Appendix Table A6, this Equation (S9) serves as the denominator for all level-2 class-specific R-squared measures for the L1L2MIX model.

Derivation of the *kh*-class-combination-specific model-implied outcome variance for the multilevel mixture regression model with classes at both levels (L1L2MIX). The denominator for the *kh*-class-combination-specific R-squared measures (i.e., the model-implied within-class-combination-*kh* outcome variance) for the L1L2MIX is derived as follows

$$\begin{aligned}\text{var}_{ij|kh}(y_{ij|c_{ij}=k,d_j=h}) &= \text{var}_{ij|kh}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(kh)} + \mathbf{x}_{j}^{b\prime}\boldsymbol{\gamma}^{b(kh)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_j + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(kh)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(kh)} + \boldsymbol{\gamma}^{b(kh)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(kh)} + tr(\mathbf{T}^{h}\boldsymbol{\Phi}^{r}) + \tau_{00}^{h} + \theta^{kh}\end{aligned} \tag{S15}$$

This Equation (S15) differs from the expression in Equation (S8) in that both *k* and *h* are held constant, and thus there is no across-class variance and there is no marginalization across either *k* or *h*. As shown in (print) Appendix Table A7, this Equation (S15) serves as the denominator for all *kh*-class-combination-specific R-squared measures for the L1L2MIX model.

Derivation of the total model-implied outcome variance for the multilevel regression mixture model with classes only at level-1 (L1MIX). For the L1MIX (manuscript Equation (2)), the total model-implied outcome variance expression from Equation (S8) simplifies to

$$\begin{aligned}\text{var}(y_{ij|c_{ij}=k}) &= \text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(k)} + \mathbf{x}_{j}^{b\prime}\boldsymbol{\gamma}^{b(k)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_j + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(\cdot)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\cdot)} + tr(\boldsymbol{\Omega}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{b(\cdot)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(\cdot)} + tr(\boldsymbol{\Omega}^{b}\boldsymbol{\Phi}^{b}) + \mathbf{m}^{b\prime}\boldsymbol{\Omega}^{b}\mathbf{m}^{b} + tr(\mathbf{T}\boldsymbol{\Phi}^{r}) + \tau_{00} + \theta^{\cdot}\end{aligned} \tag{S16}$$

Notation and terms from Equation (S16) are defined in (print) Appendix Table A1. Note that this expression in Equation (S16) is derived using the same steps as above for the L1L2MIX with classes at both levels, however, for the L1MIX model with only level-1 classes these steps exclude computations involving variation across level-2 class; hence, this expression in Equation (S16) does not include variance attributable to v_1^h, v_2^h, or m^h. As shown in (print) Appendix Table A2, this Equation (S16) serves as the denominator for all total R-squared measures for the L1MIX model.

Derivation of the level-1 class specific model-implied outcome variance for the multilevel regression mixture model with classes only at level-1 (L1MIX). The model-implied level-1-class-specific outcome variance for the L1MIX model is given as

$$\begin{aligned}\text{varr}_{ij|k}(y_{ij|c_{ij}=k}) &= \text{var}_{ij|k}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(k)} + \mathbf{x}_{j}^{b\prime}\boldsymbol{\gamma}^{b(k)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_j + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(k)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(k)} + \boldsymbol{\gamma}^{b(k)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(k)} + tr(\mathbf{T}\boldsymbol{\Phi}^{r}) + \tau_{00} + \theta^{k}\end{aligned} \tag{S17}$$

This expression in Equation (S17) differs from the total outcome variance expression in Equation (S16) in that *k* is held constant, and thus there is no variance attributable to v_1^k, v_2^k, or m^k, and there is no marginalization across *k*. As shown in (print) Appendix Table A3, this Equation (S17) serves as the denominator for all class-specific R-squared measures for the L1MIX model.

Derivation of the total model-implied outcome variance for the multilevel regression mixture model with classes only at level-2 (L2MIX). For the L2MIX model (manuscript Equation (3)), the total model-implied outcome variance expression in Equation (S8) simplifies to

$$\begin{aligned}\text{var}(y_{ij|d_j=h}) &= \text{var}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(h)} + \mathbf{x}_{j}^{b\prime}\boldsymbol{\gamma}^{b(h)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_{j} + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(\cdot)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(\cdot)} + tr(\mathbf{Z}^{w}\boldsymbol{\Phi}^{w}) + \boldsymbol{\gamma}^{b(\cdot)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(\cdot)} + tr(\mathbf{Z}^{b}\boldsymbol{\Phi}^{b}) + \mathbf{m}^{b\prime}\mathbf{Z}^{b}\mathbf{m}^{b} + tr(\mathbf{T}^{\cdot}\boldsymbol{\Phi}^{r}) + \tau_{00}^{\cdot} + \theta^{\cdot}\end{aligned} \tag{S18}$$

See Appendix Table A1 for definitions of notation and terms used in Equation (S18). The expression in Equation (S18) is derived using the same steps as above for the L1L2MIX with classes at both levels, however, for the L2MIX the steps exclude any computation involving variation across level-1 class; hence, Equation (S18) does not include variance attributable to v_1^k, v_2^k, or m^k. As shown in (print) Appendix Table A2, this Equation (S18) serves as the denominator for all total R-squared measures for the L2MIX model.

Derivation of the level-2-class-specific outcome variance for the multilevel regression mixture model with classes only at level-2 (L2MIX). The model-implied level-2-class-specific outcome variance for the L2MIX model is given as

$$\begin{aligned}\text{var}_{ij|h}(y_{ij|d_j=h}) &= \text{var}_{ij|h}(\mathbf{x}_{ij}^{w\prime}\boldsymbol{\gamma}^{w(h)} + \mathbf{x}_{j}^{b\prime}\boldsymbol{\gamma}^{b(h)} + \mathbf{w}_{ij}^{\prime}\mathbf{u}_{j} + \varepsilon_{ij}) \\ &= \boldsymbol{\gamma}^{w(h)\prime}\boldsymbol{\Phi}^{w}\boldsymbol{\gamma}^{w(h)} + \boldsymbol{\gamma}^{b(h)\prime}\boldsymbol{\Phi}^{b}\boldsymbol{\gamma}^{b(h)} + tr(\mathbf{T}^{h}\boldsymbol{\Phi}^{r}) + \tau_{00}^{h} + \theta^{h}\end{aligned} \tag{S19}$$

This expression differs from the total outcome variance expression in Equation (S18) in that h is held constant, and thus there is no variance attributable to v_1^h, v_2^h, or m^h, nor do any terms involve averaging over level-2 class. As shown in (print) Appendix Table A3, this Equation (S19) serves as the denominator for all class-specific R-squared measures for the L2MIX model.

Online Supplement Part II: Point estimates and standard error results from fitted H=2 L2MIX empirical example model predicting worker productivity

Parameter	Estimate (Standard Error)
Level-2 (Organization-level) Latent Class *h*=1	
Slope of organization-average worker satisfaction (organization-level predictor)	1.798 (2.985)
Slope of proportion female workers in the organization (organization-level predictor)	-15.372 (32.303)
Slope of worker satisfaction—fixed component (worker-level predictor)	11.335 (0.218)*
Slope of worker experience—fixed component (worker-level predictor)	6.640 (2.348)*
Intercept	195.433 (29.295)
Variance of random slope for worker experience	2.825 (1.915)†
Variance of random intercept	3330.087 (2410.855)†
Covariance of random intercept, random slope of worker experience	12.729 (123.093)
Residual variance	1623.040 (53.085)*†
Level-2 (Organization-level) Latent class *h*=2	
Slope of organization-average worker satisfaction (organization-level predictor)	0.143 (4.497)
Slope of proportion female workers in the organization (organization-level predictor)	-7.114 (65.054)
Slope of worker satisfaction—fixed component (worker-level predictor)	12.402 (0.214)*
Slope of worker experience—fixed component (worker-level predictor)	12.972 (2.621)*
Intercept	322.304 (55.527)*
Variance of random slope for experience	1.349 (1.823)
Variance of random intercept	3053.015 (1677.846)†
Covariance of random intercept, random slope of worker experience	15.820 (132.184)*
Residual variance	1860.907 (44.725)*†

Notes. *$p<.05$. †=z-tests of variance components are conservative; to address this, here we used the alpha correction approach of Fitzmaurice, Laird and Ware (2004), which has been shown to perform similarly to the mixture reference distribution approach (Ke & Wang, 2015).

Online Supplement Part III. Point estimates and standard error results from fitted K=3 H=2 L1L2MIX empirical example model predicting math achievement

Parameter	Estimate (Standard Error)
Student-level/ School-level Class-Combination *k*=1, *h*=1,	
Slope of Catholic (school-level predictor)	4.628 (1.114)*
Slope of Private (school-level predictor)	3.916 (1.638)*
Slope of per_advanced degree (school-level predictor)	0.217 (0.015)*
Slope of school-average ses (school-level predictor)	-3.478 (0.387)*
Slope of female—fixed component (student-level predictor)	0.466(0.362)
Slope of ses—fixed component (student-level predictor)	0.618 (0.017)*
Intercept	5.975 (1.179)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430) *†
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	13.164 (1.029)* †
Student-level/ School-level Class-Combination *k*=2, *h*=1	
Slope of Catholic (school-level predictor)	4.719 (2.137)*
Slope of Private (school-level predictor)	1.093 (2.286)
Slope of per_advanced degree (school-level predictor)	0.183 (0.039)*
Slope of school-average ses (school-level predictor)	-1.379 (0.774)
Slope of female—fixed component (student-level predictor)	-1.689 (1.056)
Slope of ses—fixed component (student-level predictor)	2.852 (0.042)*
Intercept	19.281 (2.131)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430)* †
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	88.728 (1.948)* †
Student-level/ School-level Class-Combination *k*=3, *h*=1	
Slope of Catholic (school-level predictor)	-27.821 (0.569)*
Slope of Private (school-level predictor)	20.227 (0.549)*
Slope of per_advanced degree (school-level predictor)	-0.374 (0.005)*
Slope of school-average ses (school-level predictor)	9.088 (0.187)*
Slope of female—fixed component (student-level predictor)	-3.766 (0.466)*
Slope of ses—fixed component (student-level predictor)	-0.257 (0.002)*
Intercept	77.043 (0.603)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430)* †
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	0.292 (0.058)* †

Student-level/ School-level Class-Combination *k*=1, *h*=2	
Slope of Catholic (school-level predictor)	4.852 (0.676)*
Slope of Private (school-level predictor)	20.244 (1.051)*
Slope of per_advanced degree (school-level predictor)	0.004 (0.008)
Slope of school-average ses (school-level predictor)	-1.814 (0.383)*
Slope of female—fixed component (student-level predictor)	-0.045 (0.209)
Slope of ses—fixed component (student-level predictor)	1.259 (0.043)*
Intercept	31.918 (0.569)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430)* †
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	13.164 (1.029)* †
Student-level/ School-level Class-Combination *k*=2, *h*=2	
Slope of Catholic (school-level predictor)	2.419 (0.715)*
Slope of Private (school-level predictor)	-8.000 (1.644)*
Slope of per_advanced degree (school-level predictor)	0.019 (0.013)
Slope of school-average ses (school-level predictor)	4.463 (0.679)*
Slope of female—fixed component (student-level predictor)	-0.755 (0.248)*
Slope of ses—fixed component (student-level predictor)	6.071 (0.204)*
Intercept	46.050 (0.820)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430)* †
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	88.728 (1.948)* †
Student-level/ School-level Class-Combination *k*=3, *h*=2	
Slope of Catholic (school-level predictor)	2.932 (0.264)*
Slope of Private (school-level predictor)	-42.566 (0.267)*
Slope of per_advanced degree (school-level predictor)	0.001 (0.004)
Slope of school-average ses (school-level predictor)	-5.134 (0.072)*
Slope of female—fixed component (student-level predictor)	-0.627 (0.139)*
Slope of ses—fixed component (student-level predictor)	-0.202 (0.002)*
Intercept	107.535 (0.295)*
Variance of random slope for female	0.970 (0.284)* †
Variance of random slope for ses	0.423 (0.089)* †
Variance of random intercept	156.478 (37.430)* †
Covariance of random intercept, random slope female	-3.427 (1.007)*
Covariance of random intercept, random slope ses	-4.732 (1.371)*
Covariance of random slope ses, random slope female	0.103 (0.092)
Student-level residual variance	0.292 (0.058)* †

Notes. *$p<.05$. †=z-tests of variance components are conservative; to address this, here we used the alpha correction approach of Fitzmaurice, Laird and Ware (2004), which has been shown to perform similarly to the mixture reference distribution approach (Ke & Wang, 2015).

Online Supplement Part IV: *reRegMixR2* R function and description

***reRegMixR2* R function Description:**

We provide an R function *reRegMixR2* that reads in random effects multilevel regression mixture model parameter estimates and outputs all single-source R-squared measures for that model. This R function is also being incorporated into the existing *r2mlm* R package (Shaw, Rights, Sterba, & Flake, 2020). That is, when the L1MIX (manuscript Equation (2)) or L2MIX (manuscript Equation (3)) are fit, single-source R-squared measures in manuscript Table 1 are outputted. When the L1L2MIX (manuscript Equation (5)) is fit, single-source R-squared measures in manuscript Table 3 are outputted. For reasons described in the manuscript, all level-1 predictors are assumed to be cluster-mean-centered.

Any number of level-1 classes and/or level-2 classes is supported along with random intercepts and/or slopes (including the special cases in which K=1 and/or H=1 and/or there are no random effects). When fitting a L1L2MIX, it is possible to have different numbers of level-1 classes for different level-2 classes. For instance, suppose in h=1 we have K=3 but in h=2 we have K=2. When entering parameter estimates for this model using the *reRegMixR2* function, the researcher would enter K=3 sets of estimates for each level-2 class, however for h=2 two of these sets of estimates would be identical.

***reRegMixR2* R function Input:**

data – Data set with rows denoting observations and columns denoting variables.

within_covs – Vector of numbers corresponding to the columns in the data set that represent the level-1 predictors used in the model

between_covs – Vector of numbers corresponding to the columns in the data set that represent the level-2 predictors used in the model

random_covs – Vector of numbers corresponding to the columns in the data set that represent the level-1 predictors with random slopes in the model

H – Number of level-2 classes.

K – Number of level-1 classes.

gamma_ws – Vector of coefficient estimates for all class-combination-specific (or class-specific) slopes of level-1 predictors, to be entered in the following order: (1) all intercepts going in order of increasing k (level-1 class) then increasing h (level-2 class) (e.g., k=1,h=1; k=2,h=1; k=1,h=2; k=2,h=2); (2) all slopes for each class-combination (classes increasing as in (1)), e.g., xslope1_k1h1, xslope2_k1h1, xslope1_k2h1, xslope2_k2h1, etc.). If coefficients are constrained equal across certain classes, then the same estimates would be entered for those classes.

gamma_bs – Vector of coefficient estimates for all class-combination-specific (or class-specific) intercepts and slopes of level-2 predictors (entered in the order of classes described in *gamma_ws* above)

Taus – List of random effect covariance matrices, entered in order of increasing h (level-2 class)

resvar – Vector of class-combination-specific (or class-specific) residual variance estimates (entered in the order of classes described in *intslopes* above) *mcwi* – Vector of level-1 class multinomial intercept estimates, entered in order of increasing *k*, with 0 entered for *K*

mcws – Vector of multinomial slopes of *k* on *h estimates* (entered in the order of classes described in *intslopes* above, with 0 entered for every *k=K* and *h=H*)

mcbi – Vector of level-2 class multinomial intercept estimates, entered in order of increasing *h*, with 0 entered for *H*

***reRegMixR2* R function Syntax:**

```
reRegMixR2 <-
  function(data,within_covs,between_covs,random_covs,gamma_ws,gamma_bs,Taus,resvars,H,K,mcwi,mcws,mcbi){

    ##read in intercepts and slopes
    if(is.null(within_covs)==F) gamma_ws <- aperm(array(data=gamma_ws,dim=c(K,H,length(within_covs))),perm=c(2,1,3))
    gamma_bs <- aperm(array(data=gamma_bs,dim=c(K,H,length(between_covs)+1)),perm=c(2,1,3))

    ##read in residual variances
    residual_var <- matrix(data=resvars,H,K,byrow=TRUE)

    ##read in multinomials
    #L1 class intercepts
    mcw <- matrix(data=mcwi,1,K)
    mcw[1,K] <- 0

    #L2 class on L1 class slopes
    ms <- matrix(data=mcws,H,K,byrow=TRUE)
    ms[H,1:K] <- 0
    ms[1:H,K] <- 0

    #L2 class intercepts
    mcb <- matrix(data=mcbi,H,1)
    mcb[H,1] <- 0

    ##compute probabilities of class membership
    #denominator for p_cbcw for each cbcw
    den_cbcw <- matrix(NA,H,K)
    for (i in 1:H)
    {
      for (j in 1:K)
      {
        den_cbcw[i,j] <- sum(exp(mcw[1,1:K]+ms[i,1:K]))
      }
    }

    #prob of cw given cb
    prob_cwgivencb <- matrix(NA,H,K)
    for (i in 1:H)
    {
      for (j in 1:K)
      {
        prob_cwgivencb[i,j] <- exp(mcw[1,j]+ms[i,j]) / den_cbcw[i,j]
      }
    }

    #marginal prob of cb
    prob_cb <- matrix(NA,H,1)
    for (i in 1:H)
    {
      prob_cb[i,1] <- exp(mcb[i,1])/ sum(exp(mcb[1:H,1]))
    }

    #class combination prob
    prob_cbcw <- matrix(NA,H,K)
    for (i in 1:H)
    {
```

```
  for (j in 1:K)
  {
   prob_cbcw[i,j] <- prob_cb[i,1]*prob_cwgivencb[i,j]
  }
}

#compute marginal L2 class parameters
if(is.null(within_covs)==F){
margL2params_w <- array(NA,c(H,1,length(within_covs)))
for (i in 1:H)
{
  for (j in 1:(length(within_covs)))
  {
   margL2params_w[i,1,j] <- sum(prob_cwgivencb[i,1:K]*gamma_ws[i,1:K,j])
  }
}
}

margL2params_b <- array(NA,c(H,1,length(between_covs)+1))
for (i in 1:H)
{
  for (j in 1:c(length(between_covs)+1))
  {
   margL2params_b[i,1,j] <- sum(prob_cwgivencb[i,1:K]*gamma_bs[i,1:K,j])
  }
}

##compute gamma_w_dotdot
if(is.null(within_covs)==F){
margparams_w <- matrix(NA,length(within_covs),1)
for (i in 1:c(length(within_covs)))
{
  margparams_w[i,1] <- sum(prob_cb*margL2params_w[,,i])
}
}

##compute gamma_b_dotdot
if(is.null(between_covs)==F){
margparams_b <- matrix(NA,length(between_covs)+1,1)
for (i in 1:c(length(between_covs)+1))
{
  margparams_b[i,1] <- sum(prob_cb*margL2params_b[,,i])
}
}

##compute phi_W
if(is.null(within_covs)==F){
phi_w <- cov(data[,within_covs])
}

##compute omega_w
if(is.null(within_covs)==F){
omega_w <- matrix(NA,length(within_covs),length(within_covs))
for (i in 1:c(length(within_covs)))
{
  for (j in 1:c(length(within_covs)))
  {
   omega_w[i,j] <- sum(prob_cbcw*((gamma_ws[,,i]-margL2params_w[,,i])*
                                   (gamma_ws[,,j]-margL2params_w[,,j])))
  }
}
}

##compute omega_b
omega_b <- matrix(NA,length(between_covs)+1,length(between_covs)+1)
for (i in 1:(length(between_covs)+1))
{
  for (j in 1:(length(between_covs)+1))
  {
   omega_b[i,j] <- sum(prob_cbcw*((gamma_bs[,,i]-margL2params_b[,,i])*
```

```
                              (gamma_bs[,,j]-margL2params_b[,,j])))
     }
   }

   ##compute zeta_w
   if(is.null(within_covs)==F){
   zeta_w <- matrix(NA,length(within_covs),length(within_covs))
   for (i in 1:length(within_covs))
   {
     for (j in 1:length(within_covs))
     {
       zeta_w[i,j] <- sum(prob_cb*margL2params_w[,1,i]*margL2params_w[,1,j])-
sum(prob_cb*margL2params_w[,1,i])*sum(prob_cb*margL2params_w[,1,j])
     }
   }
   }

   ##compute zeta_b
   zeta_b <- matrix(NA,length(between_covs)+1,length(between_covs)+1)
   for (i in 1:(length(between_covs)+1))
   {
     for (j in 1:(length(between_covs)+1))
     {
       zeta_b[i,j] <- sum(prob_cb*margL2params_b[,1,i]*margL2params_b[,1,j])-
sum(prob_cb*margL2params_b[,1,i])*sum(prob_cb*margL2params_b[,1,j])
     }
   }

   ##compute phi_b
   if(is.null(between_covs)==F){
   phi_b <- cov(cbind(1,data[,between_covs]))
   }

   ##compute m_b
   m_b <- matrix(colMeans(cbind(1,data[,between_covs])),ncol=1)

   ##compute Tau_dot
   for(i in seq(length(Taus)))
   {
     if (i == 1) Tau_dot <- prob_cb[i]*Taus[[i]]
     if (i > 1) Tau_dot <- Tau_dot + prob_cb[i]*Taus[[i]]
   }

   ##compute tau_00_dot
   tau00_dot <- Tau_dot[1,1]

   ##compute phi_r
   if(is.null(random_covs)==F){
   phi_r <- cov(cbind(1,data[,random_covs]))
   }

   ##compute theta_dotdot
   margresvar <- sum(prob_cbcw*resvars)

   ##compute total variance
   if(is.null(within_covs)==F){
   f_1 <- t(margparams_w)%*%phi_w%*%margparams_w
   v_1_k <- tr(omega_w%*%phi_w)
   v_1_h <- tr(zeta_w%*%phi_w)
   }else{f_1 <-v_1_k <-v_1_h <-0}

   if(is.null(between_covs)==F){
   f_2 <- t(margparams_b)%*%phi_b%*%margparams_b
   v_2_k <- tr(omega_b%*%phi_b)
   v_2_h <- tr(zeta_b%*%phi_b)
   }else{f_2 <-v_2_k <-v_2_h <-0}

   m_k <- t(m_b)%*%omega_b%*%m_b
   m_h <- t(m_b)%*%zeta_b%*%m_b

   if(is.null(random_covs)==F){
   v_1_r <- tr(Tau_dot%*%phi_r)
```

```
  }else{v_1_r<-0}

  m_r <- tau00_dot
  sig <- margresvar

  totalvar <- f_1+f_2+v_1_k+v_1_h+v_2_k+v_2_h+m_k+m_h+v_1_r+m_r+sig
  R2_f1_t <-  f_1/totalvar
  R2_f2_t <-  f_2/totalvar
  R2_v1_k_t <-  v_1_k/totalvar
  R2_v1_h_t <-  v_1_h/totalvar
  R2_v2_k_t <-  v_2_k/totalvar
  R2_v2_h_t <-  v_2_h/totalvar
  R2_m_k_t <-  m_k/totalvar
  R2_m_h_t <-  m_h/totalvar
  R2_v1_r_t <-  v_1_r/totalvar
  R2_m_r_t <-  m_r/totalvar
  sig_t <- sig/totalvar

  ####level-2 decomposition

  gamma_w_doth <-matrix(NA,length(within_covs),H)
  gamma_b_doth <- matrix(NA,length(between_covs)+1,H)
  sig_doth <- NA

  for(i in seq(H)){
   if(is.null(within_covs)==F) gamma_w_doth[,i] <- matrix(c(margL2params_w[i,,]),ncol=1)
   if(is.null(between_covs)==F)gamma_b_doth[,i] <- matrix(c(margL2params_b[i,,]),ncol=1)
  sig_doth[i] <- sum(prob_cwgivencb[i,1:K]*residual_var[i,1:K])
  }

  omega_wh <- list()
  if(is.null(within_covs)==F)gamma_w_temp <- aperm(array(data=gamma_ws,dim=c(K,H,length(within_covs))),perm=c(2,1,3))
  omega_wh <- aperm(array(NA,dim=c(length(within_covs),H,length(within_covs))),perm=c(2,1,3))
  for(q in 1:H){
   for (i in 1:(length(within_covs)))
   {
    for (j in 1:(length(within_covs)))
    {
     if(is.null(within_covs)==F) omega_wh[q,i,j] <- sum(prob_cwgivencb[q,1:K]*gamma_w_temp[q,1:K,i]*gamma_w_temp[q,1:K,j])-
sum(prob_cwgivencb[q,1:K]*gamma_w_temp[q,1:K,i])*sum(prob_cwgivencb[q,1:K]*gamma_w_temp[q,1:K,j])
    }
   }
  }

  omega_bh <- list()
  gamma_b_temp <- aperm(array(data=gamma_bs,dim=c(K,H,length(between_covs)+1)),perm=c(2,1,3))
  omega_bh <- aperm(array(NA,dim=c(length(between_covs)+1,H,length(between_covs)+1)),perm=c(2,1,3))
  for(q in 1:H){
   for (i in 1:(length(between_covs)+1))
   {
    for (j in 1:(length(between_covs)+1))
    {
     omega_bh[q,i,j] <- sum(prob_cwgivencb[q,1:K]*gamma_b_temp[q,1:K,i]*gamma_b_temp[q,1:K,j])-
sum(prob_cwgivencb[q,1:K]*gamma_b_temp[q,1:K,i])*sum(prob_cwgivencb[q,1:K]*gamma_b_temp[q,1:K,j])
    }
   }
  }

  R2_f1_h <-R2_f2_h<-R2_v1_k_h<-R2_v2_k_h<-R2_m_k_h<-R2_v1_r_h<-R2_m_r_h<-sig_h<- NA
  for(i in seq(H)){
   if(is.null(within_covs)==F){
    f_1_h <- t(gamma_w_doth[,i])%*%phi_w%*%gamma_w_doth[,i]
    v_1_k_h <- tr(omega_wh[i,,]%*%phi_w)
   }else{f_1_h<-v_1_k_h<-0}

   if(is.null(between_covs)==F){
    f_2_h <- t(gamma_b_doth[,i])%*%phi_b%*%gamma_b_doth[,i]
    v_2_k_h <- tr(omega_bh[i,,]%*%phi_b)
   }else{f_2_h<-v_2_k_h<-0}

   m_k_h <- t(m_b)%*%omega_bh[i,,]%*%m_b

   if(is.null(random_covs)==F){
```

```
      v_1_r_h <- tr(Taus[[i]]%*%phi_r)
     }else{v_1_r_h<-0}

    m_r_h <- Taus[[i]][1,1]
    sigh <- sig_doth[i]

    totalvar_h <- f_1_h+f_2_h+v_1_k_h+v_2_k_h+m_k_h+v_1_r_h+m_r_h+sigh
    R2_f1_h[i] <- f_1_h/totalvar_h
    R2_f2_h[i] <- f_2_h/totalvar_h
    R2_v1_k_h[i] <- v_1_k_h/totalvar_h
    R2_v2_k_h[i] <- v_2_k_h/totalvar_h
    R2_m_k_h[i] <- m_k_h/totalvar_h
    R2_v1_r_h[i] <- v_1_r_h/totalvar_h
    R2_m_r_h[i] <- m_r_h/totalvar_h
    sig_h[i] <- sigh/totalvar_h
   }

   ###class-combination-specific decomposition

   R2_f1_kh<-R2_f2_kh<-R2_v_kh<-R2_m_kh<-sig_kh<-matrix(NA,H,K)
   for(i in seq(H)){
     for(j in seq(K)){
       if(is.null(within_covs)==F){f_1_kh <- gamma_ws[i,j,]%*%phi_w%*%gamma_ws[i,j,]}else{f_1_kh<-0}
       if(is.null(between_covs)==F){f_2_kh <- gamma_bs[i,j,]%*%phi_b%*%gamma_bs[i,j,]}else{f_2_kh<-0}
       if(is.null(random_covs)==F){v_kh <- tr(Taus[[i]]%*%phi_r)}else{v_kh<-0}
      m_kh <- Taus[[i]][1,1]
      sigkh <- residual_var[i,j]

      totalvar_kh <- f_1_kh+f_2_kh+v_kh+m_kh+sigkh

      R2_f1_kh[i,j] <- f_1_kh/totalvar_kh
      R2_f2_kh[i,j] <-f_2_kh/totalvar_kh
      R2_v_kh[i,j] <-v_kh/totalvar_kh
      R2_m_kh[i,j] <-m_kh/totalvar_kh
      sig_kh[i,j] <-sigkh/totalvar_kh
     }
   }

   ##results tables

   if(H>1&&K>1){
   totalR2_table <- matrix(c(R2_f1_t,R2_f2_t,R2_v1_k_t,R2_v1_h_t,R2_v2_k_t,
                       R2_v2_h_t,R2_m_k_t,R2_m_h_t,R2_v1_r_t,R2_m_r_t,sig_t),ncol=1)
   rownames(totalR2_table) <- c("f1","f2","v1_k","v1_h","v2_k","v2_h","m_k","m_h","v1_r","m_r","residual")
   colnames(totalR2_table) <- c(" ")

   level2R2_table <- rbind(c(R2_f1_h),c(R2_f2_h),c(R2_v1_k_h),c(R2_v2_k_h),
                           c(R2_m_k_h),c(R2_v1_r_h),c(R2_m_r_h),c(sig_h))

   rownames(level2R2_table) <- c("f1","f2","v1_k","v2_k","m_k","v1_r","m_r","residual")
   colnames(level2R2_table) <- c(paste("h =",1:H))

   rownames(R2_f1_kh)<-rownames(R2_f2_kh)<-  rownames(R2_v_kh) <-
    rownames(R2_m_kh)<-rownames(sig_kh)<-c(paste("k =",1:K))

  colnames(R2_f1_kh)<-colnames(R2_f2_kh)<-  colnames(R2_v_kh) <-
    colnames(R2_m_kh)<-colnames(sig_kh)<-c(paste("h =",1:H))

   Output <- list(totalR2_table,level2R2_table,
              R2_f1_kh,R2_f2_kh,R2_v_kh,R2_m_kh,sig_kh)
   names(Output) <- c("total decomposition","L2 class decomposition","class comb. R2_f1","class comb. R2_f2","class comb. R2_v","class comb. R2_m","class comb. prop. residual")
   }

   if(H>1&&K==1){
     totalR2_table <- matrix(c(R2_f1_t,R2_f2_t,R2_v1_h_t,
                       R2_v2_h_t,R2_m_h_t,R2_v1_r_t,R2_m_r_t,sig_t),ncol=1)
     rownames(totalR2_table) <- c("f1","f2","v1_h","v2_h","m_h","v1_r","m_r","residual")
     colnames(totalR2_table) <- c(" ")

     level2R2_table <- rbind(c(R2_f1_h),c(R2_f2_h),
                       c(R2_v1_r_h),c(R2_m_r_h),c(sig_h))
```

```
   rownames(level2R2_table) <- c("f1","f2","v1_r","m_r","residual")
   colnames(level2R2_table) <- c(paste("h =",1:H))

   Output <- list(totalR2_table,level2R2_table)
   names(Output) <- c("total decomposition","L2 class decomposition")
  }

  if(H==1&&K>1){
   totalR2_table <- matrix(c(R2_f1_t,R2_f2_t,R2_v1_k_t,R2_v2_k_t,
                        R2_m_k_t,R2_v1_r_t,R2_m_r_t,sig_t),ncol=1)

   rownames(totalR2_table) <- c("f1","f2","v1_k","v2_k","m_k","v1_r","m_r","residual")
   colnames(totalR2_table) <- c(" ")


   level1R2_table <- rbind(c(R2_f1_kh),c(R2_f2_kh),c(R2_v_kh),c(R2_m_kh),c(sig_kh))


   rownames(level1R2_table) <- c("f1","f2","v1_r","m_r","residual")
   colnames(level1R2_table) <- c(paste("k =",1:K))

   Output <- list(totalR2_table,level1R2_table)
   names(Output) <- c("total decomposition","L1 class decomposition")
  }

  if(H==1&&K==1){
   totalR2_table <- matrix(c(R2_f1_t,R2_f2_t,
                        R2_v1_r_t,R2_m_r_t,sig_t),ncol=1)
   rownames(totalR2_table) <- c("f1","f2","v1_r","m_r","residual")
   colnames(totalR2_table) <- c(" ")


   Output <- list(totalR2_table,level1R2_table)
   names(Output) <- c("totalR2")
  }
  return(Output)
 }
```

reRegMixR2 Illustrative input:

```
#NOTE: #the estimates in the input represent hypothetical results for a H=3, K=3 class model with 2 leve1-1 predictors and 2 level-2
#predictors. In practice a user would have previously obtained these input estimates by fitting their model in multilevel mixture software
#additionally, the input consists of simulated covariate data, whereas in practice a user would read-in their actual covariate data

data<-mvrnorm(100,c(0,0,1,1),matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1),4,4))

reRegMixR2(data=data,
        within_covs=c(1,2),
        between_covs=c(3,4),
        random_covs=c(1,2),
        gamma_ws=c(1,1,2,2,1,2,1,1,2,
             8,4,1,.5,4,2,5,1,5),
        gamma_bs=c(1,1,2,2,1,1,2,2,1,
             1,1,2,2,1,1,2,2,1,
             1,1,2,2,1,1,2,2,1),
        Taus=list(matrix(c(4,0,0,
                    0,1,0,
                    0,0,1),3,3),
              matrix(c(6,0,0,
                    0,2,0,
                    0,0,1),3,3),
              matrix(c(4,0,0,
                    0,4,0,
                    0,0,1),3,3)),
        resvars=c(1,1,1,1,2,4,1,8,8),
        H=3,K=3,
        mcwi=c(1,1,0),
        mcws=c(1,1,0,2,2,0,0,0,0),
        mcbi=c(1,1,0))
```