

Write Anything Here

SwitZ Shichien Kylin-OIO

2025 年 3 月 13 日

1 Porter Robinson Discography

1.1 Worlds(2014)

- Divinity(feat. Amy Millan)
- Sad Machine
- Years of War(feat. Breanne Duren and Sean Caskey)
- Flicker
- Fresh Static Snow
- Hollowheart(2024)(feat. Amy Millan)
- Polygon Dust(feat. Lemaitre)
- Hear the Bells(feat. Imaginary Cities)
- Natural Light
- Lionhearted(feat. Urban Cone)
- Sea of Voices
- Fellow Feeling
- Goodbye to a World
- Shepherdess/She Heals Everything(2021)

1.2 Nurture(2021)

- Lifelike
- Look at the Sky
- Get Your Wish
- Wind Tempos
- Musician
- Do-re-mi-fa-so-la-ti-do
- Mother
- Dullscythe
- Sweet Time
- Mirror
- Something Comforting
- Blossom
- Unfold(feat. Totally Enormous Extinct Dinosaurs)
- Trying to Feel Alive
- Fullmoon Lullaby(feat. Wednesday Campanella)

1.3 SMILE!:D(2024)

- Knock Yourself Out XD
- Cheerleader
- Russian Roulette
- Perfect Pinterest Garden
- Year of the Cup
- Kistune Maison Freestyle

- Easier to Love You
- Mona Lisa
- Is There Really No Happiness?
- Everthing to Me

2 冲突示例

我是来捣蛋的。

3 Switz 的表格

Column 1	Column 2	Column 3
<i>Left-aligned text</i>	<i>Centered text</i>	<i>Right-aligned text</i>
Another left-aligned text	Another centered text	Another right-aligned text

表 1: SwitZ 的表格

如表 1所示，这是 SwitZ 的表格。Hollowheart 如 2 所示，第三行第三列已改为代码字体。

4 Kylin-OIO 的表格

Column 1	Column 2	Column 3
<i>Left aligned text</i>	<i>Centered text</i>	<i>Right aligned text</i>
Another left aligned text	Another centered text	Another right aligned text

表 2: 三线表示例

如 2 所示，第三行第三列已改为代码字体。

5 数学代码

行内公式使用一个美元符号包裹，如 $a^2 + b^2 = c^2$ 。

行间公式使用两个美元符号包裹，如

$$\int_0^1 x^2 dx = \frac{1}{3}$$

在数学公式中，使用下划线表示下标，使用脱字符表示上标，如 a_1^2 。

如果只有一个字符作为上标或下标，可以省略大括号，如 a^2 。

如果有多个字符要作为上标或者下标，需要大括号：Deralive^{Switz}_{Kylin-OIO}

你会发现我把文本都用 `\text{}` 包裹起来了，是因为在美元符号的公式环境中，不用包裹的文本会被解释为数学符号。

有一些符号是不能随意打出来的，需要转义。例如反斜杠、百分号（因为反斜杠是 LaTeX 中的命令前缀，百分号是 LaTeX 中的注释，这些特殊的符号都需要转义。转义的方法是在前面加上一个反斜杠，像我这样写%）

但是反斜杠的输入方法最特别，要使用一个命令来输出这个符号：`\`

LaTeX 的数学公式是有表格的，除了用 GPT 之外，还可以自己查表：

L^AT_EX Mathematical Symbols

The more unusual symbols are not defined in base L^AT_EX (NFSS) and require `\usepackage{amssymb}`

1 Greek and Hebrew letters

α	<code>\alpha</code>	κ	<code>\kappa</code>	ψ	<code>\psi</code>	F	<code>\digamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>
β	<code>\beta</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ε	<code>\varepsilon</code>	Γ	<code>\Gamma</code>	Υ	<code>\Upsilon</code>
χ	<code>\chi</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	\varkappa	<code>\varkappa</code>	Λ	<code>\Lambda</code>	Ξ	<code>\Xi</code>
δ	<code>\delta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>	φ	<code>\varphi</code>	Ω	<code>\Omega</code>		
ϵ	<code>\epsilon</code>	o	<code>o</code>	θ	<code>\theta</code>	ϖ	<code>\varpi</code>	Φ	<code>\Phi</code>	\aleph	<code>\aleph</code>
η	<code>\eta</code>	ω	<code>\omega</code>	v	<code>\upsilon</code>	ϱ	<code>\varrho</code>	Π	<code>\Pi</code>	\beth	<code>\beth</code>
γ	<code>\gamma</code>	ϕ	<code>\phi</code>	ξ	<code>\xi</code>	ς	<code>\varsigma</code>	Ψ	<code>\Psi</code>	\daleth	<code>\daleth</code>
ι	<code>\iota</code>	π	<code>\pi</code>	ζ	<code>\zeta</code>	ϑ	<code>\vartheta</code>	Σ	<code>\Sigma</code>	\gimel	<code>\gimel</code>

2 L^AT_EX math constructs

$\frac{abc}{xyz}$	<code>\frac{abc}{xyz}</code>	\overline{abc}	<code>\overline{abc}</code>	\overrightarrow{abc}	<code>\overrightarrow{abc}</code>
f'	<code>f'</code>	\underline{abc}	<code>\underline{abc}</code>	\overleftarrow{abc}	<code>\overleftarrow{abc}</code>
\sqrt{abc}	<code>\sqrt{abc}</code>	\widehat{abc}	<code>\widehat{abc}</code>	\overbrace{abc}	<code>\overbrace{abc}</code>
$\sqrt[n]{abc}$	<code>\sqrt[n]{abc}</code>	\widetilde{abc}	<code>\widetilde{abc}</code>	\underbrace{abc}	<code>\underbrace{abc}</code>

3 Delimiters

$ $	<code> </code>	$\{$	<code>\{</code>	\lfloor	<code>\lfloor</code>	$/$	<code>/</code>	\Uparrow	<code>\Uparrow</code>	\llcorner	<code>\llcorner</code>
\backslash	<code>\backslash</code>	$\}$	<code>\}</code>	\rfloor	<code>\rfloor</code>	\backslash	<code>\backslash</code>	\uparrow	<code>\uparrow</code>	\lrcorner	<code>\lrcorner</code>
$\ $	<code>\ </code>	\langle	<code>\langle</code>	\lceil	<code>\lceil</code>	\lceil	<code>\lceil</code>	\Downarrow	<code>\Downarrow</code>	\ulcorner	<code>\ulcorner</code>
$\ $	<code>\ </code>	\rangle	<code>\rangle</code>	\rceil	<code>\rceil</code>	\rfloor	<code>\rfloor</code>	\downarrow	<code>\downarrow</code>	\urcorner	<code>\urcorner</code>

Use the pair `\lefts1` and `\right2` to match height of delimiters s_1 and s_2 to the height of their contents, e.g.,
`\left| expr \right|` `\left\{ expr \right\}` `\left\Vert expr \right\Vert`.

4 Variable-sized symbols (displayed formulae show larger version)

\sum	<code>\sum</code>	\int	<code>\int</code>	\biguplus	<code>\biguplus</code>	\bigoplus	<code>\bigoplus</code>	\bigvee	<code>\bigvee</code>
\prod	<code>\prod</code>	\oint	<code>\oint</code>	\bigcap	<code>\bigcap</code>	\bigotimes	<code>\bigotimes</code>	\bigwedge	<code>\bigwedge</code>
\coprod	<code>\coprod</code>	\iint	<code>\iint</code>	\bigcup	<code>\bigcup</code>	\bigodot	<code>\bigodot</code>	\bigsqcup	<code>\bigsqcup</code>

5 Standard Function Names

Function names should appear in Roman, not Italic, e.g.,

Correct: `\tan(at-n\pi) \longrightarrow \tan(at-n\pi)`
 Incorrect: `\tan(at-n\pi) \longrightarrow \tan(at-n\pi)`

\arccos	<code>\arccos</code>	\arcsin	<code>\arcsin</code>	\arctan	<code>\arctan</code>	\arg	<code>\arg</code>
\cos	<code>\cos</code>	\cosh	<code>\cosh</code>	\cot	<code>\cot</code>	\coth	<code>\coth</code>
\csc	<code>\csc</code>	\deg	<code>\deg</code>	\det	<code>\det</code>	\dim	<code>\dim</code>
\exp	<code>\exp</code>	\gcd	<code>\gcd</code>	\hom	<code>\hom</code>	\inf	<code>\inf</code>
\ker	<code>\ker</code>	\lg	<code>\lg</code>	\lim	<code>\lim</code>	\liminf	<code>\liminf</code>
\limsup	<code>\limsup</code>	\ln	<code>\ln</code>	\log	<code>\log</code>	\max	<code>\max</code>
\min	<code>\min</code>	\Pr	<code>\Pr</code>	\sec	<code>\sec</code>	\sin	<code>\sin</code>
\sinh	<code>\sinh</code>	\sup	<code>\sup</code>	\tan	<code>\tan</code>	\tanh	<code>\tanh</code>

图 1: Lat_EX 数学符号表

6 SwitZ 的数学公式和图片

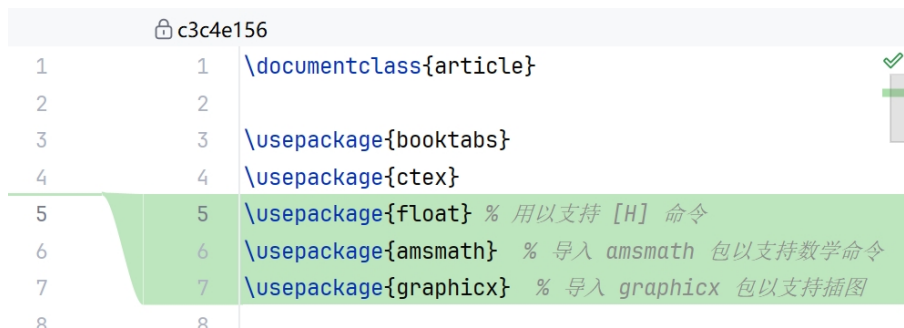


图 2: SwitZ 的图片

如图 2所示，这是 SwitZ 的图片。

下方是 SwitZ 的数学公式：

公式定义：综合考虑组件版本回退的影响，计算其成本：

$$\text{Cost}(i) = \alpha_i \times \sum_{r_{i,j} \in R_i} \left(\alpha_j \times \frac{\Delta X_i}{X_i} (10 - |X_i - X_j|) + \alpha_j \times \frac{\Delta Y_i}{Y_i} (10 - \beta |Y_i - Y_j|) \right)$$

其中：

- α_i : 组件 c_{pi} 的权重。
- R_i : 组件 c_{pi} 的依赖集合。
- $\Delta X_i = |X_i^* - X_i|$: 主版本号的差值。

7 Kylin-OIO 的数学公式和图片